

August 2014

### GENERAL DESCRIPTION

The HI-3210 from Holt Integrated Circuits is a single chip CMOS data management IC capable of managing, storing and forwarding avionics data messages between eight ARINC 429 receive channels and four ARINC 429 transmit channels.

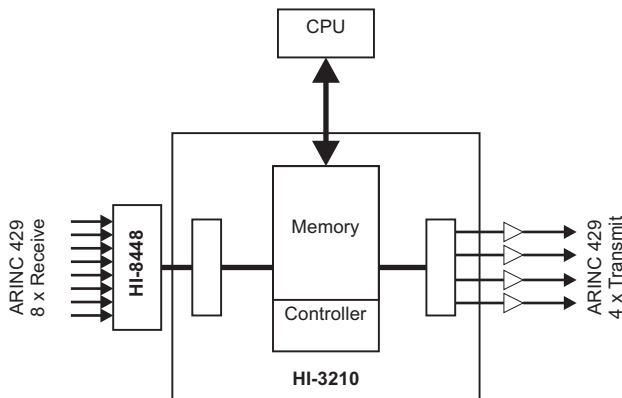
The ARINC 429 buses may be operated independently, allowing a host CPU to send and receive data on multiple buses, or the HI-3210 can be programmed to automatically re-format, re-label, re-packetize and re-transmit data from ARINC 429 receive buses to ARINC 429 transmit buses.

A 32K x 8 on-board memory allows received data to be logically organized and automatically updated as new ARINC 429 labels are received.

An auto-initialization feature allows configuration information to be up-loaded from an external EEPROM on reset to facilitate rapid start-up or operation without a host CPU.

The HI-3210 interfaces directly with Holt's HI-8448 octal ARINC 429 receiver IC and HI-8592 or HI-8596 ARINC 429 line drivers.

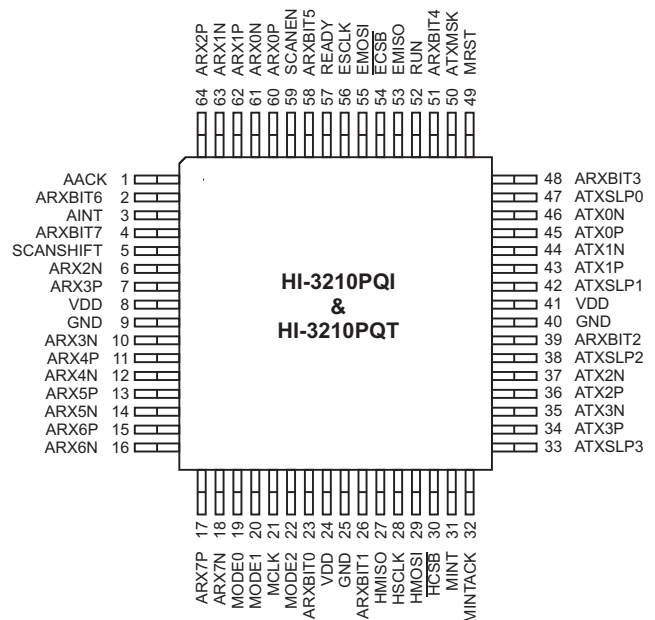
### APPLICATION



### FEATURES

- Eight ARINC 429 Receive channels
- Four ARINC 429 Transmit channels
- 32KB on chip user-configurable data storage memory
- Programmable received data filtering
- Programmable transmission schedulers for periodic ARINC 429 broadcasting
- SPI Host CPU interface
- Auto-initialization feature allows power-on configuration or independent operation without CPU

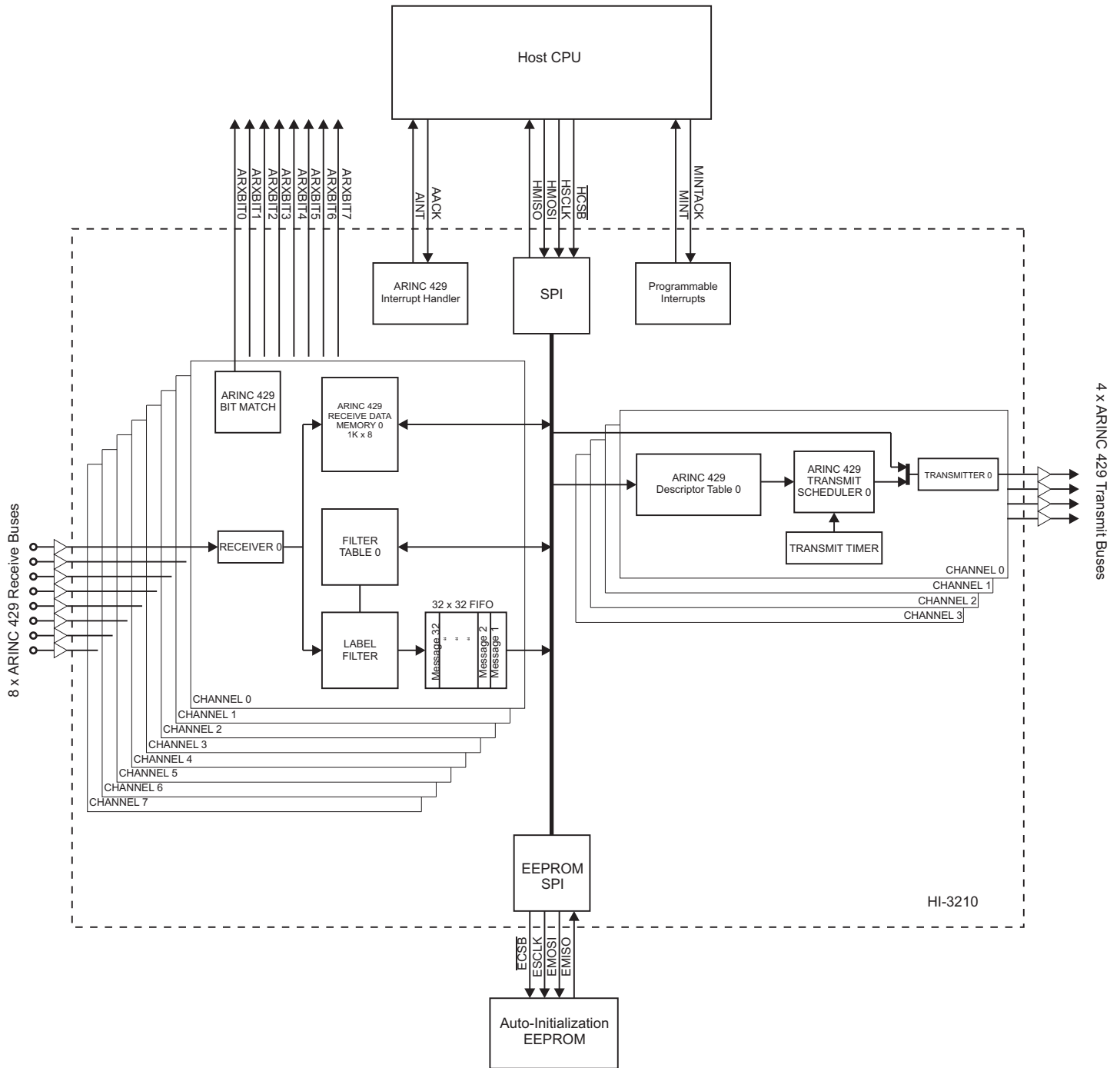
### PIN CONFIGURATION



#### 64 - Pin Plastic Quad Flat Pack (PQFP)

(See ordering information for additional pin configurations)

BLOCK DIAGRAM



# APPLICATION OVERVIEW

The HI-3210 is a flexible device for managing ARINC 429 communications and data storage in many avionics applications. The device architecture centers around a 32K x 8 static RAM used for data storage, data filtering tables and table-driven transmission schedulers. Once configured, the device can operate autonomously without a host CPU, negating the need for software development or DO-178 certification. Configuration data may be uploaded into the device from an external EEPROM, following system reset.

The device supports up to eight ARINC 429 receive channels. Received data is stored in on-chip RAM organized by channel number and label. The data table continually updates as new labels arrive. Programmable interrupts and filters alert the host subsystem to labels of interest.

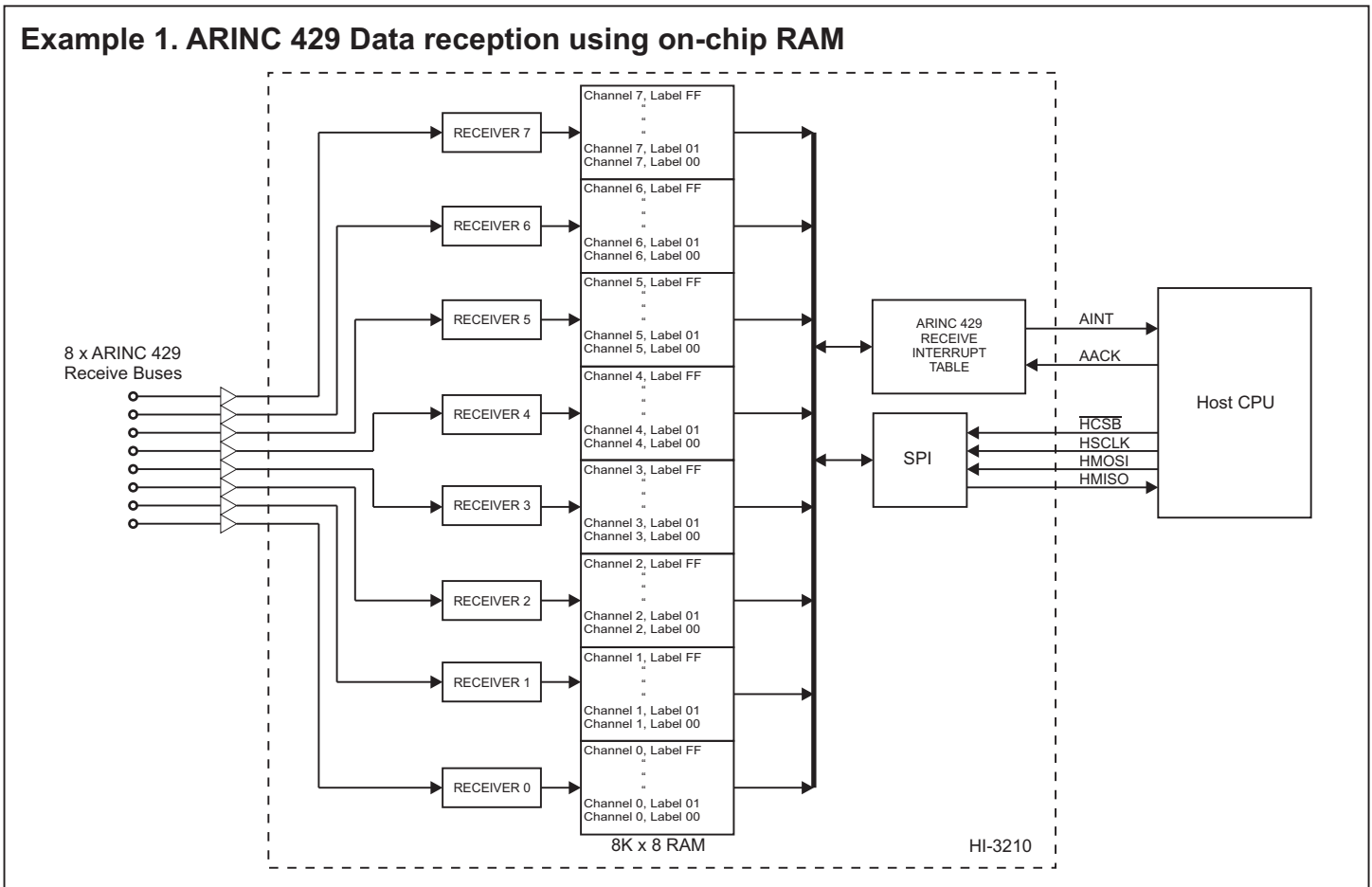
Each ARINC 429 receive channel also includes a 32 message deep FIFO allowing selected label data to be queued for subsequent host access.

The HI-3210 includes four independent ARINC 429 transmit channels. Transmission may be controlled entirely by an external CPU, or autonomously by programming one or more of the four on-chip ARINC 429 transmit schedulers. These allow periodic transmission to occur without CPU. Source data for transmission may be selected from RAM based tables of constants and / or from the received channel data. Powerful options exist for constructing ARINC 429 labels as well as controlling their timing and conditional transmission.

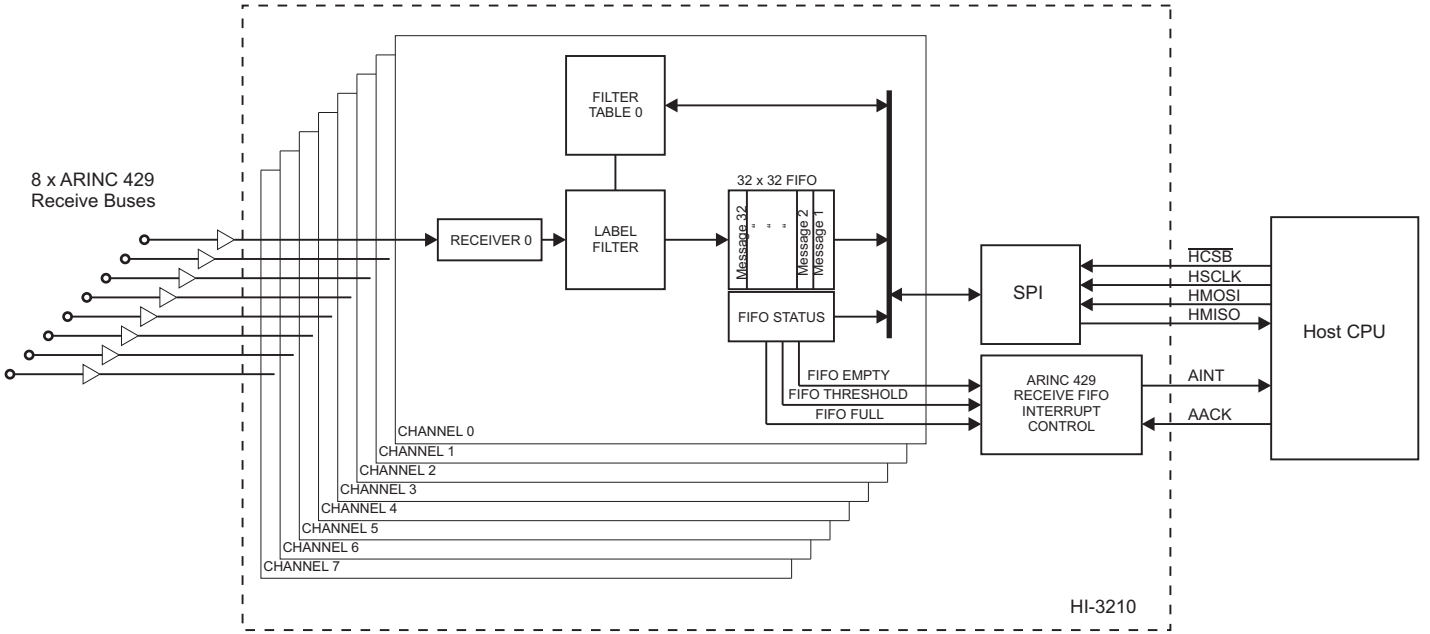
Even when running under the control of schedulers, the host CPU may insert new labels for transmission at will.

The following examples show five possible configurations of how the HI-3210 may be used:

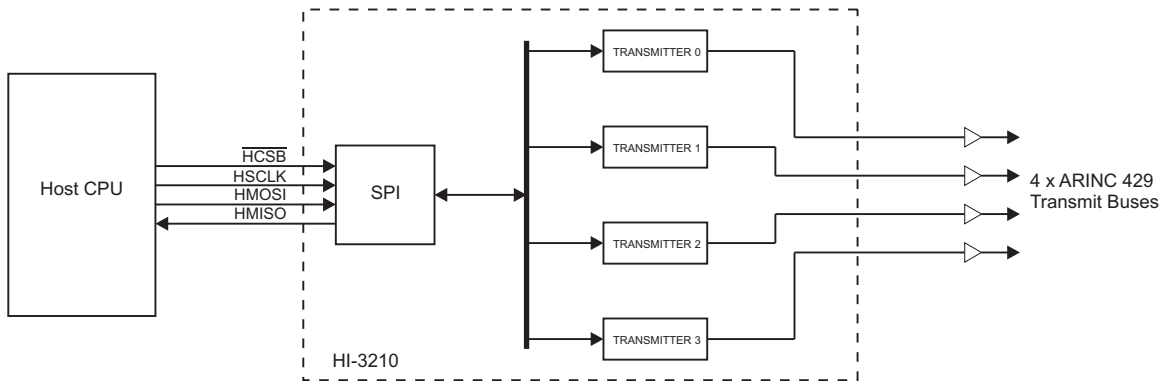
## Example 1. ARINC 429 Data reception using on-chip RAM



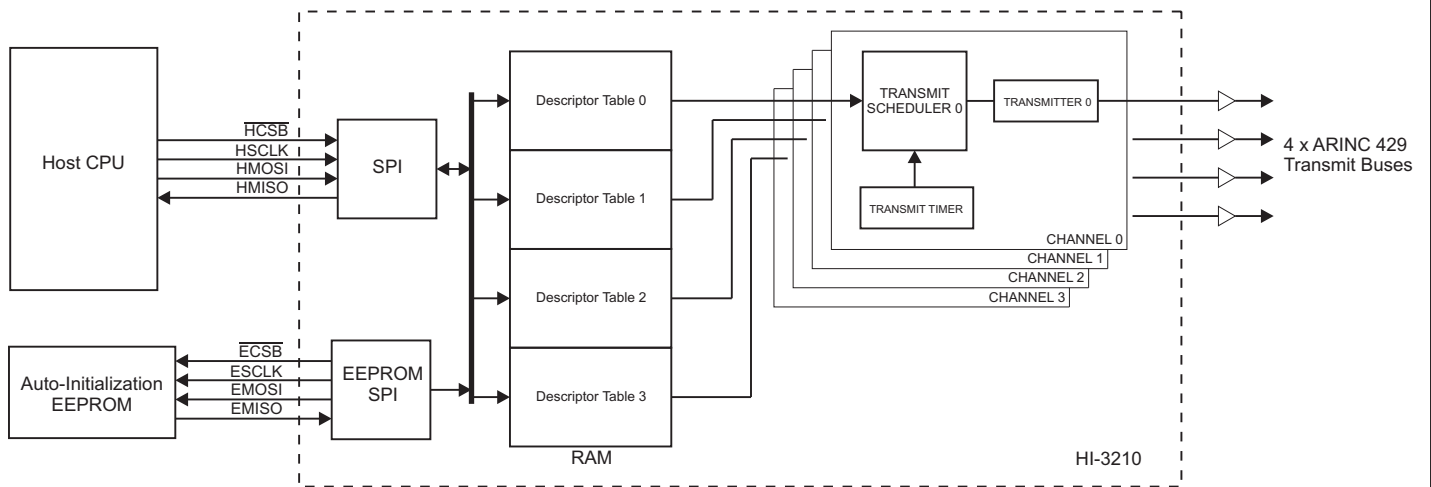
**Example 2. ARINC 429 Data reception using on-chip filters and FIFOs**



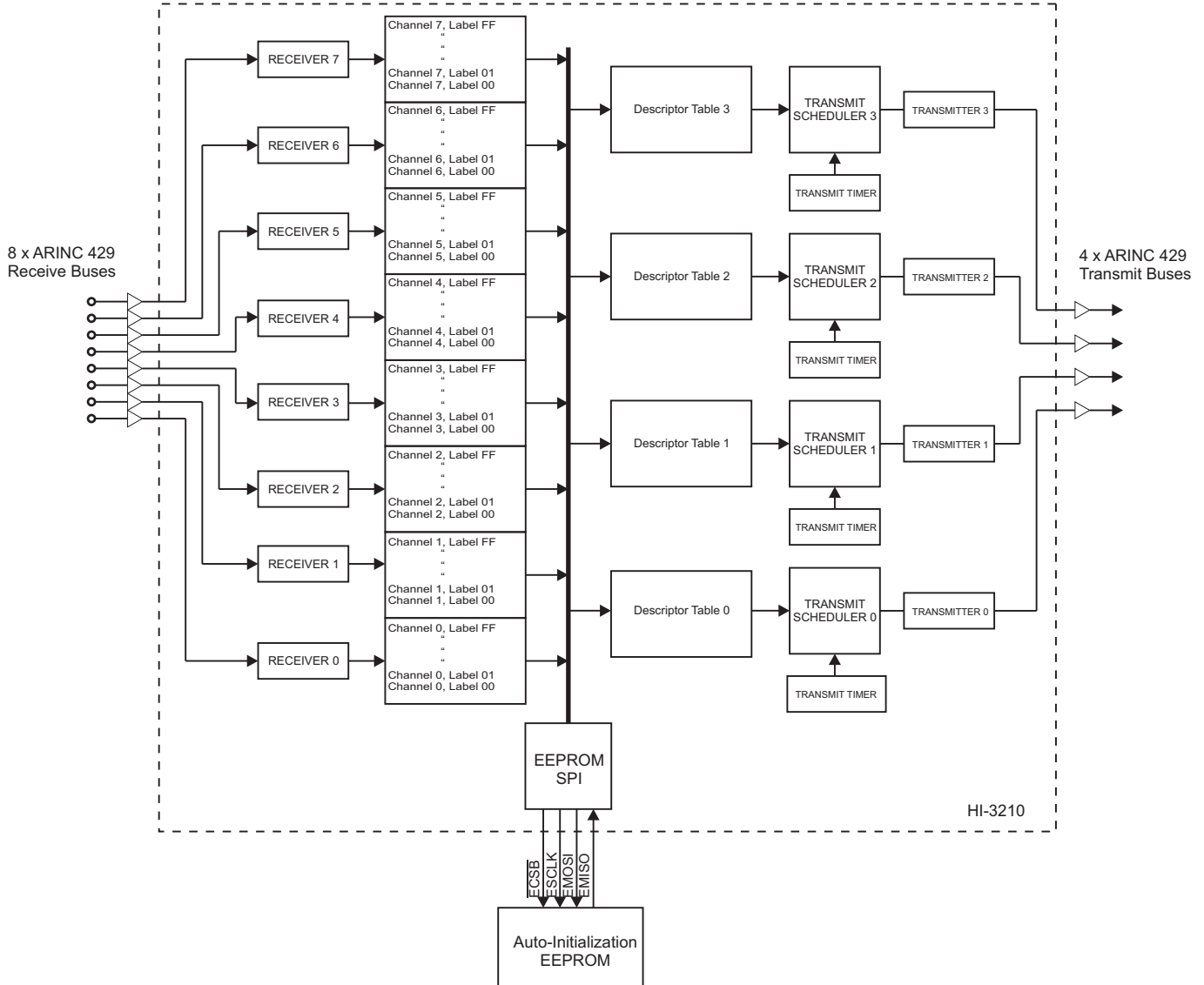
**Example 3. ARINC 429 Data transmission directly from CPU**



**Example 4. ARINC 429 Data transmission using on-chip schedulers**



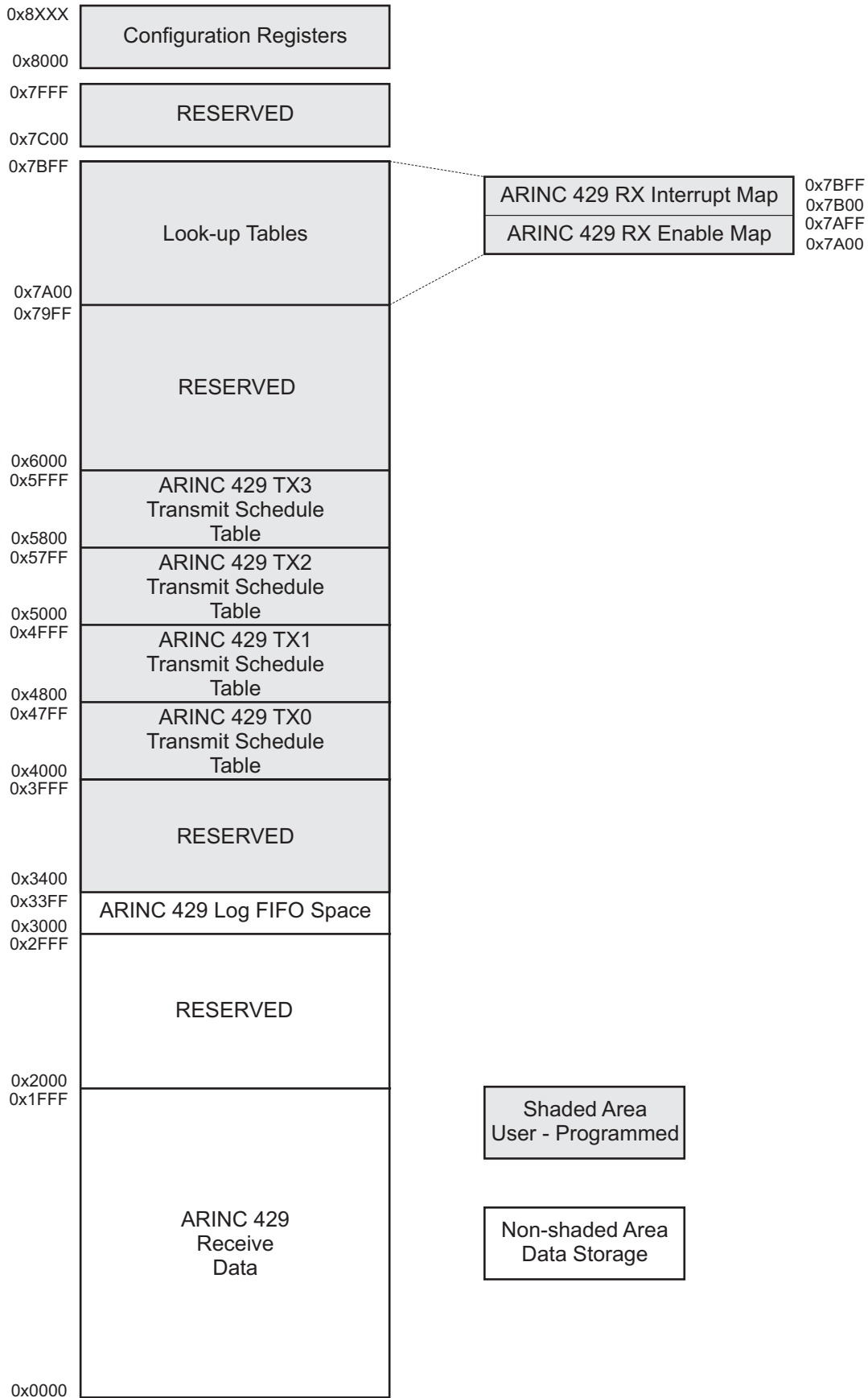
**Example 5. Autonomous ARINC 429 Data Concentrator / Repeater**



## PIN DESCRIPTIONS

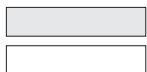
SIGNAL	FUNCTION	DESCRIPTION
AACK	INPUT	ARINC 429 Receiver Interrupt Acknowledge
AINT	OUTPUT	ARINC 429 Receiver Interrupt
ARX0N	INPUT	ARINC 429 Rx negative data input for channel 0
ARX0P	INPUT	ARINC 429 Rx positive data input for channel 0
ARX1N	INPUT	ARINC 429 Rx negative data input for channel 1
ARX1P	INPUT	ARINC 429 Rx positive data input for channel 1
ARX2N	INPUT	ARINC 429 Rx negative data input for channel 2
ARX2P	INPUT	ARINC 429 Rx positive data input for channel 2
ARX3N	INPUT	ARINC 429 Rx negative data input for channel 3
ARX3P	INPUT	ARINC 429 Rx positive data input for channel 3
ARX4N	INPUT	ARINC 429 Rx negative data input for channel 4
ARX4P	INPUT	ARINC 429 Rx positive data input for channel 4
ARX5N	INPUT	ARINC 429 Rx negative data input for channel 5
ARX5P	INPUT	ARINC 429 Rx positive data input for channel 5
ARX6N	INPUT	ARINC 429 Rx negative data input for channel 6
ARX6P	INPUT	ARINC 429 Rx positive data input for channel 6
ARX7N	INPUT	ARINC 429 Rx negative data input for channel 7
ARX7P	INPUT	ARINC 429 Rx positive data input for channel 7
ARXBIT0-7	OUTPUTS	ARINC 429 received payload bit monitor pins 0 through 7
ATX0N	OUTPUT	ARINC 429 Tx channel 0 negative data output to line driver
ATX0P	OUTPUT	ARINC 429 Tx channel 0 positive data output to line driver
ATX1N	OUTPUT	ARINC 429 Tx channel 1 negative data output to line driver
ATX1P	OUTPUT	ARINC 429 Tx channel 1 positive data output to line driver
ATX2N	OUTPUT	ARINC 429 Tx channel 2 negative data output to line driver
ATX2P	OUTPUT	ARINC 429 Tx channel 2 positive data output to line driver
ATX3N	OUTPUT	ARINC 429 Tx channel 3 negative data output to line driver
ATX3P	OUTPUT	ARINC 429 Tx channel 3 positive data output to line driver
ATXMSK	INPUT	Turn off ARINC 429 Transmit pins (Holds TXnA/B pins zero)
ATXSLP0	OUTPUT	ARINC 429 Tx channel 0 data rate select output. 1 = high speed, 0 = low speed
ATXSLP1	OUTPUT	ARINC 429 Tx channel 1 data rate select output. 1 = high speed, 0 = low speed
ATXSLP2	OUTPUT	ARINC 429 Tx channel 2 data rate select output. 1 = high speed, 0 = low speed
ATXSLP3	OUTPUT	ARINC 429 Tx channel 3 data rate select output. 1 = high speed, 0 = low speed
ECSB	OUTPUT	SPI chip select for auto-initialization EEPROM
EMISO	INPUT	SPI serial data input from auto-initialization EEPROM
EMOSI	OUTPUT	SPI serial data output to auto-initialization EEPROM
ESCLK	OUTPUT	SPI clock for auto-initialization EEPROM
GND	POWER	Chip 0V supply
HCSB	INPUT	Chip select. Data is shifted into HMOSI and out of HMISO when HCSB is low
HMISO	OUTPUT	Host CPU SPI interface serial data output
HMOSI	INPUT	Host CPU SPI interface serial data input
HSCLK	INPUT	SPI Clock. Data is shifted into or out of the SPI interface using HSCLK
MCLK	INPUT	Master 48 MHZ and reference clock for ARINC 429 bus bit timing
MINT	OUTPUT	Programmable event interrupt output
MINTACK	INPUT	Programmable event interrupt acknowledge
MODE2:0	INPUTS	MODE2 through MODE0 define HI-3210 start-up and initialization mode
MRST	INPUT	Master Reset to HI-3210 Active High. See RESET AND START-UP section.
PROG (MODE0)	INPUT	Multiplexed with MODE0 pin, PROG initiates HI-3210 Auto-Initialization EEPROM programming routine
READY	OUTPUT	READY goes high when post-RESET initialization is complete
RUN	INPUT	Master enable signal for ARINC 429 transmit schedulers
SCANEN	RESERVED	Connect to GND
SCANSHIFT	RESERVED	Connect to GND
VDD	POWER	3.3V power supply

# HI-3210 MEMORY MAP



# HI-3210 REGISTER MAP

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x8000	R*	ARINC 429 Rx PENDING INTERRUPT	APIR	Defines channel(s) with pending Interrupt
0x8001	R	ARINC 429 Rx INTERRUPT ADDRESS 0	AIAR0	ARINC 429 Interrupt Vector channel 0
0x8002	R	ARINC 429 Rx INTERRUPT ADDRESS 1	AIAR1	ARINC 429 Interrupt Vector channel 1
0x8003	R	ARINC 429 Rx INTERRUPT ADDRESS 2	AIAR2	ARINC 429 Interrupt Vector channel 2
0x8004	R	ARINC 429 Rx INTERRUPT ADDRESS 3	AIAR3	ARINC 429 Interrupt Vector channel 3
0x8005	R	ARINC 429 Rx INTERRUPT ADDRESS 4	AIAR4	ARINC 429 Interrupt Vector channel 4
0x8006	R	ARINC 429 Rx INTERRUPT ADDRESS 5	AIAR5	ARINC 429 Interrupt Vector channel 5
0x8007	R	ARINC 429 Rx INTERRUPT ADDRESS 6	AIAR6	ARINC 429 Interrupt Vector channel 6
0x8008	R	ARINC 429 Rx INTERRUPT ADDRESS 7	AIAR7	ARINC 429 Interrupt Vector channel 7
0x8009	-	RESERVED		
0x800A	R*	PENDING INTERRUPT REGISTER	PIR	Indicates Interrupt type
0x800B	R	RESERVED		
0x800C	R	MUXED FIFO FLAGS	AMFF	ARINC 429 Multiplexed FIFO flags
0x800D	R	ARINC 429 TX READY BITS	ATRB	ARINC 429 Transmitter Ready flags
0x800E	R	MASTER STATUS REGISTER	MSR	Indicates HI-3200 current status
0x800F	R/W	MASTER CONTROL REGISTER	MCR	HI-3200 global configuration
0x8010	R/W	ARINC 429 RX CONTROL REGISTER 0	ARXC0	Configures ARINC 429 receive channel 0
0x8011	R/W	ARINC 429 RX CONTROL REGISTER 1	ARXC1	Configures ARINC 429 receive channel 1
0x8012	R/W	ARINC 429 RX CONTROL REGISTER 2	ARXC2	Configures ARINC 429 receive channel 2
0x8013	R/W	ARINC 429 RX CONTROL REGISTER 3	ARXC3	Configures ARINC 429 receive channel 3
0x8014	R/W	ARINC 429 RX CONTROL REGISTER 4	ARXC4	Configures ARINC 429 receive channel 4
0x8015	R/W	ARINC 429 RX CONTROL REGISTER 5	ARXC5	Configures ARINC 429 receive channel 5
0x8016	R/W	ARINC 429 RX CONTROL REGISTER 6	ARXC6	Configures ARINC 429 receive channel 6
0x8017	R/W	ARINC 429 RX CONTROL REGISTER 7	ARXC7	Configures ARINC 429 receive channel 7
0x8018	R/W	ARINC 429 TX CONTROL REGISTER 0	ATXC0	Configures ARINC 429 transmit channel 0
0x8019	R/W	ARINC 429 TX CONTROL REGISTER 1	ATXC1	Configures ARINC 429 transmit channel 1
0x801A	R/W	ARINC 429 TX CONTROL REGISTER 2	ATXC2	Configures ARINC 429 transmit channel 2
0x801B	R/W	ARINC 429 TX CONTROL REGISTER 3	ATXC3	Configures ARINC 429 transmit channel 3
0x801C	R/W	ARINC 429 TX REPETITION RATE 0	ATXRR0	Sets sequence repeat time for ARINC TX0
0x801D	R/W	ARINC 429 TX REPETITION RATE 1	ATXRR1	Sets sequence repeat time for ARINC TX1
0x801E	R/W	ARINC 429 TX REPETITION RATE 2	ATXRR2	Sets sequence repeat time for ARINC TX2
0x801F	R/W	ARINC 429 TX REPETITION RATE 3	ATXRR3	Sets sequence repeat time for ARINC TX3
0x8020	R/W	ARINC 429 Rx INTERRUPT MASK	AIMR	Enables Interrupts on AINT pin
0x8021	R/W	ARINC 429 Rx FIFO THRESHOLD VALUE	AFTV	Sets flag value for ARINC 429 Receive FIFO
0x8022	R/W	ARINC 429 LOOPBACK	ALOOP	Sets loop-back self-test mode
0x8029	R	ARINC 429 Rx FIFO FULL FLAG	AFFF	Indicates which FIFOs are full
0x802A	R	ARINC 429 Rx FIFO THRESHOLD FLAG	AFTF	Indicates which FIFOs hold > (thresh) messages
0x802B	R	ARINC 429 Rx FIFO NOT EMPTY FLAG	FFNE	Indicates which receive FIFOs hold data
0x802C	R	ARINC 429 TX SEQUENCE POINTER 0	ATXSP0	Current address of ARINC transmit sequence 0
0x802D	R	ARINC 429 TX SEQUENCE POINTER 1	ATXSP1	Current address of ARINC transmit sequence 1
0x802E	R	ARINC 429 TX SEQUENCE POINTER 2	ATXSP2	Current address of ARINC transmit sequence 2
0x802F	R	ARINC 429 TX SEQUENCE POINTER 3	ATXSP3	Current address of ARINC transmit sequence 3
0x8034	R/W	PENDING INTERRUPT ENABLE REGISTER	IMR	Enables Interrupts on INT pin
0x8035	R/W	ARINC 429 TX READY INT ENABLE	ATRIE	Enables ARINC 429 TX Ready Interrupts



Fast Access Registers

Memory mapped register access only

\* Register is cleared when read (auto clear)



ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x805F	R/W	PINS ARXBIT[7:0]	ARXBIT	Values of pins ARXBIT[7:0]
0x8060	R/W	PIN ARXBIT0 CONFIG REG 1	ARX0CR1	ARINC 429 bit Monitor 0 channel & bit select
0x8061	R/W	PIN ARXBIT0 CONFIG REG 2	ARX0CR2	ARINC 429 bit Monitor 0 label select
0x8062	R/W	PIN ARXBIT1 CONFIG REG 1	ARX1CR1	ARINC 429 bit Monitor 1 channel & bit select
0x8063	R/W	PIN ARXBIT1 CONFIG REG 2	ARX1CR2	ARINC 429 bit Monitor 1 label select
0x8064	R/W	PIN ARXBIT2 CONFIG REG 1	ARX2CR1	ARINC 429 bit Monitor 2 channel & bit select
0x8065	R/W	PIN ARXBIT2 CONFIG REG 2	ARX2CR2	ARINC 429 bit Monitor 2 label select
0x8066	R/W	PIN ARXBIT3 CONFIG REG 1	ARX3CR1	ARINC 429 bit Monitor 3 channel & bit select
0x8067	R/W	PIN ARXBIT3 CONFIG REG 2	ARX3CR2	ARINC 429 bit Monitor 3 label select
0x8068	R/W	PIN ARXBIT4 CONFIG REG 1	ARX4CR1	ARINC 429 bit Monitor 4 channel & bit select
0x8069	R/W	PIN ARXBIT4 CONFIG REG 2	ARX4CR2	ARINC 429 bit Monitor 4 label select
0x806A	R/W	PIN ARXBIT5 CONFIG REG 1	ARX5CR1	ARINC 429 bit Monitor 5 channel & bit select
0x806B	R/W	PIN ARXBIT5 CONFIG REG 2	ARX5CR2	ARINC 429 bit Monitor 5 label select
0x806C	R/W	PIN ARXBIT6 CONFIG REG 1	ARX6CR1	ARINC 429 bit Monitor 6 channel & bit select
0x806D	R/W	PIN ARXBIT6 CONFIG REG 2	ARX6CR2	ARINC 429 bit Monitor 6 label select
0x806E	R/W	PIN ARXBIT7 CONFIG REG 1	ARX7CR1	ARINC 429 bit Monitor 7 channel & bit select
0x806F	R/W	PIN ARXBIT7 CONFIG REG 2	ARX7CR2	ARINC 429 bit Monitor 7 label select
0x8070	R	BIST CONTROL/STATUS	BISTS	Built-In Self-Test
0x8071	R	BIST FAIL ADDRESS [7:0]	BISTFL	Low-order failing BIST memory address
0x8072	R	BIST FAIL ADDRESS [12:8]	BISTFH	High-order failing BIST memory address
0x8073	R	AUTO-INIT FAIL LS ADDRESS [7:0]	AIFL	Auto-initialization fail address (low-byte)
0x8074	R	AUTO-INIT FAIL MS ADDRESS [15:8]	AIFH	Auto-initialization fail address (high byte)

## HI-3210 SYSTEM CONFIGURATION

Starting at memory address 0x8000, the HI-3210 contains a set of registers that are used to configure the device.

The user needs only to program the HI-3210 configuration registers to completely define the full system operation.

The configuration registers are divided into three categories, as follows;

1. HI-3210 global configuration
2. ARINC 429 Receive channel configuration
3. ARINC 429 Transmit channel configuration

### HI-3210 Global Configuration

The following registers define the HI-3210 top-level configuration:

#### MASTER CONTROL REGISTER (Address 0x800F)

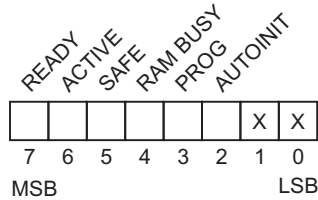


Bit	Name	R/W	Default	Description
7	A429RX	R/W	0	This bit must be set to a "1" to allow the HI-3210 to receive ARINC 429 data on any of the eight channels. If set to a zero, the HI-3210 will not respond to any ARINC 429 receive bus, regardless of the state of the ARINC 429 Receive channel Control Registers.
6	A429TX	R/W	0	This bit must be set to a "1" to allow the HI-3210 to transmit ARINC 429 data on any of the four channels. If set to a zero, the HI-3210 will not output ARINC 429 data and the ARINC 429 transmit sequencers will remain in their reset state.
5	-	R/W	0	<b>Must be '0'</b>
4	-	R/W	0	<b>Must be '0'</b>
3	AFLIP	R/W	0	When set to a "1", this bit switches the bit order of the ARINC 429 label byte in both receive and transmit channels.
2	-	R/W	0	Not Used
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

## HI-3210 Operational Status Information

The Master Status Register may be read at any time to determine the current operational state of the HI-3210:

### MASTER STATUS REGISTER (Address 0x800E)



<u>Bit</u>	<u>Name</u>	<u>R/W</u>	<u>Default</u>	<u>Description</u>
7	READY	R	0	This bit is high, when the READY output pin is high, indicating that the part is able to accept and respond to host CPU SPI commands
6	ACTIVE	R	0	This bit is high after RUN is asserted and the HI-3210 is in normal operating mode.
5	SAFE	R	0	This bit goes high when the part enters safe mode as a result of a Built-in Self-test fail or auto-initialization fail.
4	RAM BUSY	R	0	This is high during the time the RAM Integrity Check is running and RAM is clearing
3	PROG	R	0	Indicates that the HI-3210 is currently in the EEPROM programming cycle. Note that READY stays low until the cycle is complete.
2	AUTOINIT	R	0	The HI-3210 is currently loading internal memory, registers and look-up tables from the Auto-initialization EEPROM
1	-	R	0	Not used
0	-	R	0	Not Used

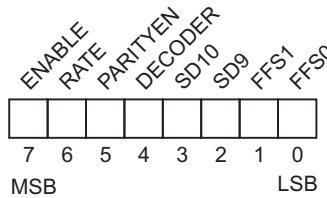
## ARINC 429 RECEIVE OPERATION

The HI-3210 can receive ARINC 429 messages from up to eight ARINC 429 receive buses. External analog line receivers handle the physical layer connection

### ARINC 429 Receive Channel Configuration

Each of the eight possible ARINC 429 Receive channels is configured using its own Control Register. Register address 0x8010 controls ARINC 429 Receive channel #0, register address 0x8011 controls channel #1 and so on. ARINC 429 Receive Control Registers may be read at any time, but can only be written when the device is in the IDLE state (RUN input = "0", READY output = "1").

#### ARINC 429 RX CONTROL REGISTER 0 - 7 (Address 0x8010 - 0x8017)



Bit	Name	R/W	Default	Description
7	ENABLE	R/W	0	This bit must be set to a "1" to enable ARINC 429 data reception on this channel.
6	RT/LO	R/W	0	Selects the ARINC 429 bit rate for the ARINC 429 receive channel. A "0" selects high-speed (100Kb/s) and a "1" selects low-speed (12.5Kb/s).
5	PARITYEN	R/W	0	When this bit is a one, the 32nd received ARINC bit is overwritten with a parity flag. The flag bit is set to a zero when the received ARINC word, including its parity bit has an odd number of ones. When PARITYEN is a zero, all 32-bits are received without parity checking.
4	DECODER	R/W	0	When DECODER is a "1", bits 9 and 10 of ARINC 429 words received on this channel must match the SD9 and SD10 bits in the register. ARINC words received that do not match the SD conditions are ignored.
3	SD10	R/W	0	If DECODER is set to a "1", then this bit must match the received ARINC word bit 10 for the word to be accepted.
2	SD9	R/W	0	If DECODER is set to a "1", then this bit must match the received ARINC word bit 9 for the word to be accepted.
1-0	FFS1:0	R/W	0	FFS1 and FFS0 define when this channel's FIFO Flag is set, as shown below. The FIFO Flag will be set in the ARINC 429 Muxed FIFO Flags Register (0x800C). Bit 0 is receive channel 0 FIFO Flag, bit 1 is receive channel 1 FIFO Flag, bit 2 is receive channel 2 FIFO Flag, etc.

FFS1	FFS0	FLAG set condition
0	0	FLAG never set
0	1	Set FLAG if FIFO NOT EMPTY bit = "1"
1	0	Set FLAG if FIFO > Threshold value
1	1	Set FLAG is FIFO FULL bit "1"

## ARINC 429 Received Data Management

The HI-3210 supports eight ARINC 429 receive buses using on-chip receivers to handle the protocol validation. The eight ARINC 429 RX Control Registers, ARXC0 - 7, define the characteristics of each receive channel.

The ARINC 429 receive function of the HI-3210 is activated by setting the A429RX bit in the Master Control Register.

When an ARINC 429 message is received by the HI-3210 on any bus, it is checked for protocol compliance. Messages with incorrect encoding are rejected.

The HI-3210 contains an 8K byte memory for storing ARINC 429 received data. The memory is organized by channel number and ARINC 429 label value. Four bytes of memory are dedicated to each channel / label to store the 32-word ARINC 429 message.

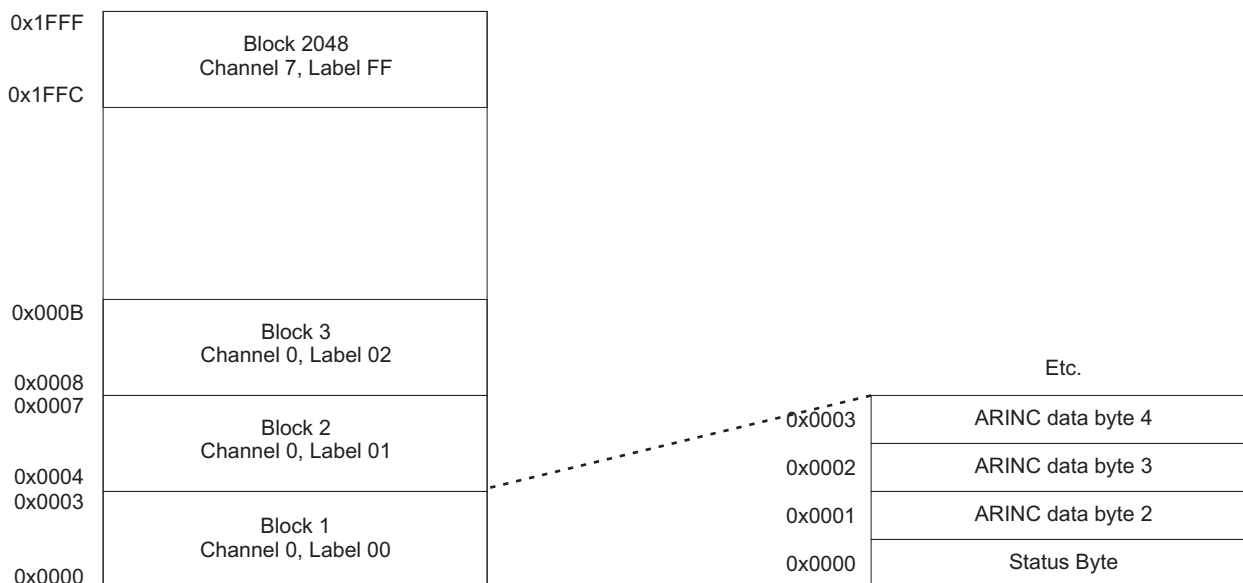
A look-up table is used to enable an interrupt on receipt of a new ARINC 429 message. Look-up table bit positions pre-loaded with a "1" will cause an Interrupt to be generated.

When a message is received that triggers an Interrupt, that channel's Interrupt bit is set in the ARINC 429 Receive Pending Interrupt Register. If this bit is unmasked in the ARINC 429 Receive Interrupt Mask Register, the AINT output pin is asserted. The label number of the ARINC 429 message causing the interrupt is loaded into that channel's ARINC 429 Receive Interrupt Address Register (AIAR0 - AIAR7).

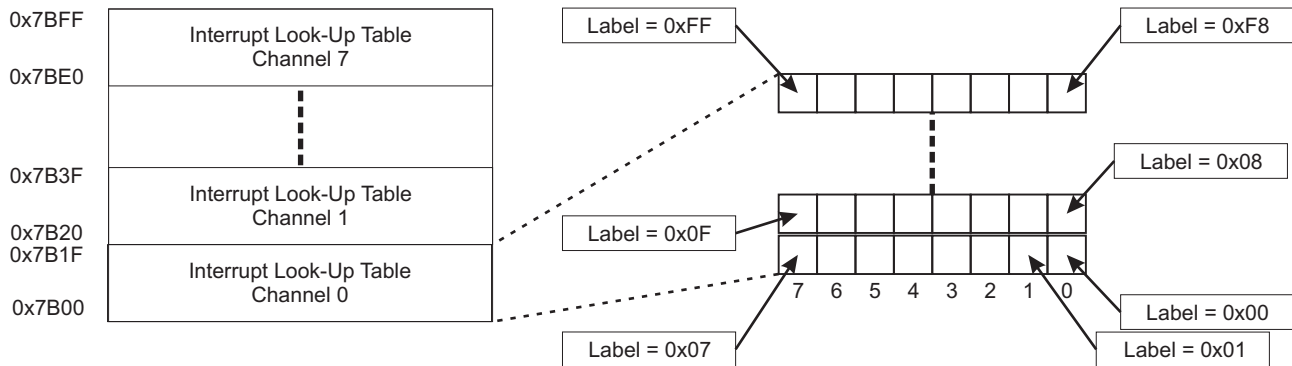
Because the ARINC Receive Memory is organized by label value, it is not necessary to store the received label value (first eight bits of the ARINC message) in the memory. Instead, the first byte is used to store a status byte.

The six active bits of the status byte are set to "1" when a new ARINC word is stored in the memory. These bits flag the ARINC word as new when the location is interrogated by the host CPU or any of the four ARINC 429 transmit schedulers.

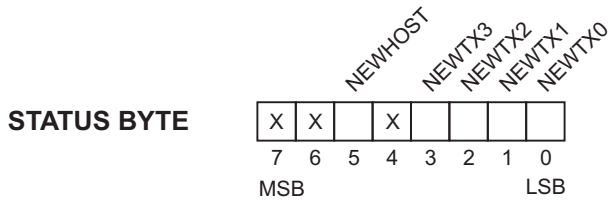
## ARINC 429 Received Data Memory Organization



**ARINC 429 Received Data Interrupt Look-Up Table**



**ARINC 429 Received Data Status Byte Definition**



Bit	Name	R/W	Default	Description
7	-	R/W	0	Not used
6	-	R/W	0	Not used
5	NEWHOST	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the host CPU executes SPI instruction 0xC0 - 0xFF to read the block.
4	-	R/W	0	Not Used
3	NEWTX3	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #3 reads any bytes from the block.
2	NEWTX2	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #2 reads any bytes from the block.
1	NEWTX1	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #1 reads any bytes from the block.
0	NEWTX0	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #0 reads any bytes from the block.

## ARINC 429 Received Data Log FIFO

A 1K x 8 block of memory located between 0x3000 and 0x33FF is reserved for a set of eight ARINC 429 received data FIFOs. There is one FIFO for each ARINC 429 received data channel. Each FIFO can hold up to 32 ARINC 429 32-bit messages.

A look-up table driven filter defines which ARINC 429 messages are stored in each FIFO. The look-up table is pre-loaded with a “1” for each bit position corresponding to a selected channel / label combination. The look-up table is located at memory address 0x7A00.

When a new ARINC 429 message is received that meets the programmed conditions for acceptance (Enable look-up table bit = “1”), it is written into the channel’s Receive Data FIFO. The contents of the FIFO may be read by the host CPU using dedicated FIFO read SPI Instructions.

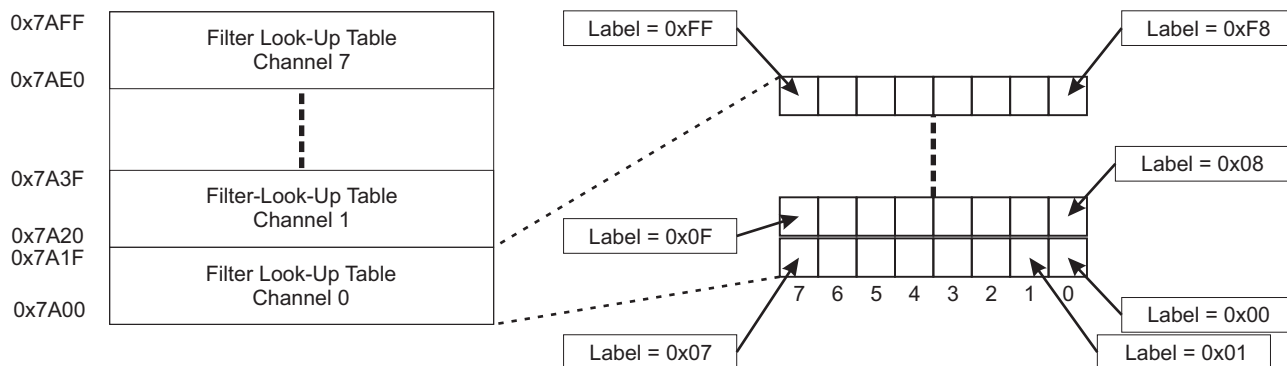
The status of each channel’s FIFOs is monitored by three FIFO status registers: FIFO NOT EMPTY, FIFO THRESHOLD, and FIFO FULL. One bit of each register reflects the current status of each FIFO.

The FIFOs are empty following Reset. All three status registers are cleared. When an ARINC 429 message is written to a FIFO, its FIFO NOT EMPTY bit is set to a “1”. When the FIFO contains more than the user-defined number of messages as programmed in the ARINC FIFO THRESHOLD VALUE register, its FIFO THRESHOLD bit is set. If the FIFO is allowed to accumulate 32 messages, its FIFO FULL bit is set. Once a FIFO is full, subsequent messages continue to be written to the FIFO, and the oldest message is lost.

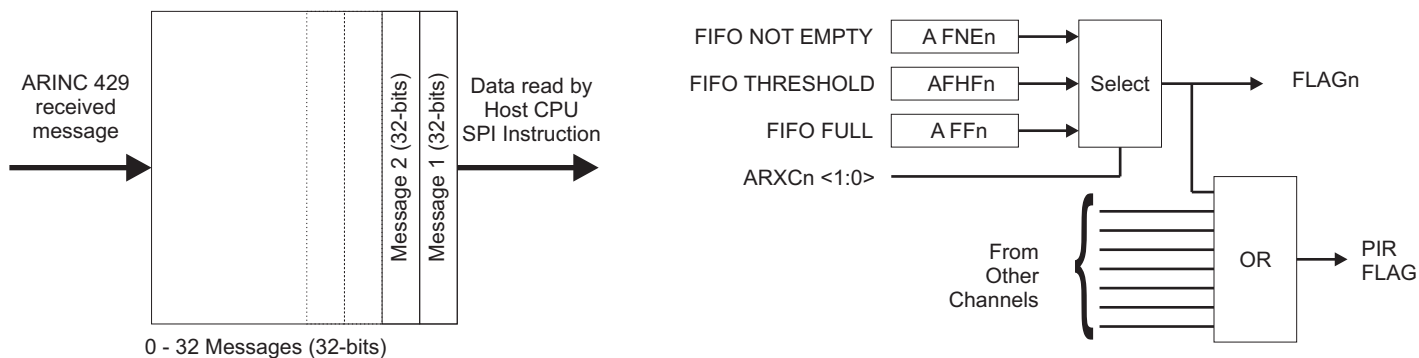
The user may generate an Interrupt by enabling one of the three FIFO status register bits to assert the FLAG bit in the Pending Interrupt Register. ARINC 429 Control Register bits 1:0 select the condition to trigger the FLAG interrupt.

The FIFO feature is particularly useful if the application wishes to accumulate sequential ARINC 429 messages of the same label value before reading them. The regular ARINC 429 receive data memory will, of course, overwrite messages of the same label value if a new message is received before the host CPU extracts the data.

## ARINC 429 Received Data Enable Look-Up Table

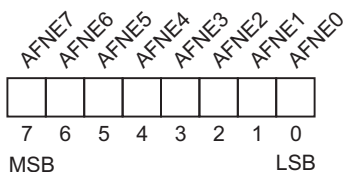


## ARINC 429 Received Data FIFO (x8)



ARINC 429 Received Data FIFO Status Registers

**FIFO NOT EMPTY REGISTER**  
(Address 0x802B)



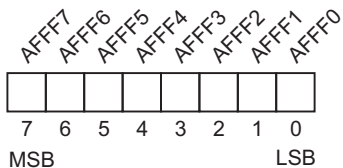
Bit	Name	R/W	Default	Description
7	AFNE7	R	0	This bit is set to “1” if FIFO #7 contains at least one ARINC 429 message
6	AFNE6	R	0	This bit is set to “1” if FIFO #6 contains at least one ARINC 429 message
5	AFNE5	R	0	This bit is set to “1” if FIFO #5 contains at least one ARINC 429 message
4	AFNE4	R	0	This bit is set to “1” if FIFO #4 contains at least one ARINC 429 message
3	AFNE3	R	0	This bit is set to “1” if FIFO #3 contains at least one ARINC 429 message
2	AFNE2	R	0	This bit is set to “1” if FIFO #2 contains at least one ARINC 429 message
1	AFNE1	R	0	This bit is set to “1” if FIFO #1 contains at least one ARINC 429 message
0	AFNE0	R	0	This bit is set to “1” if FIFO #0 contains at least one ARINC 429 message

**FIFO THRESHOLD REGISTER**  
(Address 0x802A)



Bit	Name	R/W	Default	Description
7	AFTF7	R	0	This bit is set to “1” if FIFO #7 contains > threshold number of ARINC 429 messages
6	AFTF6	R	0	This bit is set to “1” if FIFO #6 contains > threshold number of ARINC 429 messages
5	AFTF5	R	0	This bit is set to “1” if FIFO #5 contains > threshold number of ARINC 429 messages
4	AFTF4	R	0	This bit is set to “1” if FIFO #4 contains > threshold number of ARINC 429 messages
3	AFTF3	R	0	This bit is set to “1” if FIFO #3 contains > threshold number of ARINC 429 messages
2	AFTF2	R	0	This bit is set to “1” if FIFO #2 contains > threshold number of ARINC 429 messages
1	AFTF1	R	0	This bit is set to “1” if FIFO #1 contains > threshold number of ARINC 429 messages
0	AFTF0	R	0	This bit is set to “1” if FIFO #0 contains > threshold number of ARINC 429 messages

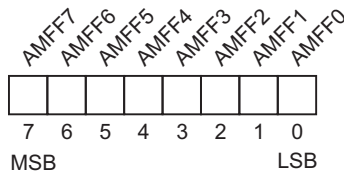
**FIFO FULL REGISTER**  
(Address 0x8029)



Bit	Name	R/W	Default	Description
7	AFFF7	R/W	0	This bit is set to “1” if FIFO #7 contains 32 ARINC 429 messages
6	AFFF6	R/W	0	This bit is set to “1” if FIFO #6 contains 32 ARINC 429 messages
5	AFFF5	R/W	0	This bit is set to “1” if FIFO #5 contains 32 ARINC 429 messages
4	AFFF4	R/W	0	This bit is set to “1” if FIFO #4 contains 32 ARINC 429 messages
3	AFFF3	R/W	0	This bit is set to “1” if FIFO #3 contains 32 ARINC 429 messages
2	AFFF2	R/W	0	This bit is set to “1” if FIFO #2 contains 32 ARINC 429 messages
1	AFFF1	R/W	0	This bit is set to “1” if FIFO #1 contains 32 ARINC 429 messages
0	AFFF0	R/W	0	This bit is set to “1” if FIFO #0 contains 32 ARINC 429 messages

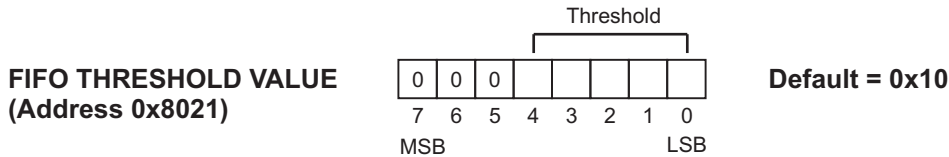


## ARINC 429 Muxed FIFO Flags Register (Address 0x800C)



Bit	Name	R/W	Default	Description
7	AMFF7	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 7. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 7
6	AMFF6	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 6. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 6
5	AMFF5	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 5. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 5
4	AMFF4	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 4. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 4
3	AMFF3	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 3. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 3
2	AMFF2	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 2. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 2
1	AMFF1	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 1. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 1
0	AMFF0	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 0. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 0

## ARINC 429 FIFO Threshold Value Register



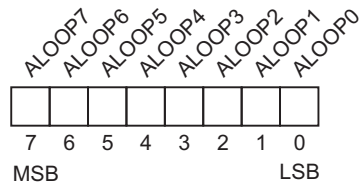
Threshold Value	Description
00000	Threshold flag is set if at least 1 message is in FIFO (Same as FIFO NOT EMPTY FLAG)
00001	Threshold flag is set if more than one message are in the FIFO
00010	Threshold flag is set if more than two messages are in the FIFO
00011	Threshold flag is set if more than three messages are in the FIFO
⋮	
10000	Threshold flag is set if more than sixteen messages are in the FIFO (default)
⋮	
11111	Threshold flag is set if 32 messages are in the FIFO (Same as FIFO FULL FLAG)

## ARINC 429 Loop-back Self-Test

The HI-3210 includes an ARINC 429 loop-back feature, which allows users to exercise the ARINC 429 transmit and receive channels for self-test purposes. The ARINC 429 Loop-Back register, ALOOP defines which receiver channels are in loop-back mode. When a “1” is programmed in the ALOOP bit position for a receiver, then its ARINC 429 bus connection to the external pins is broken and instead the input is connected to one of the four ARINC 429 transmit channels. Transmit channel 0 is connected to receive channel 0 and 1, transmit channel 1 is connected to receive channels 2 and 3, and so on.

When in loop-back mode, incoming ARINC 429 messages are ignored by the HI-3210. When running in loop-back mode the ARINC 429 transmit pins may be disabled by pulling the TXMSK input high. This prevents test messages from being output to the external ARINC 429 transmit buses.

## ARINC 429 LOOPBACK (Address 0x8022)



Bit	Name	R/W	Default	Description
7	ALOOP7	R/W	0	This bit is set to "1" to loop-back transmit channel 3 to receiver 7
6	ALOOP6	R/W	0	This bit is set to "1" to loop-back transmit channel 3 to receiver 6
5	ALOOP5	R/W	0	This bit is set to "1" to loop-back transmit channel 2 to receiver 5
4	ALOOP4	R/W	0	This bit is set to "1" to loop-back transmit channel 2 to receiver 4
3	ALOOP3	R/W	0	This bit is set to "1" to loop-back transmit channel 1 to receiver 3
2	ALOOP2	R/W	0	This bit is set to "1" to loop-back transmit channel 1 to receiver 2
1	ALOOP1	R/W	0	This bit is set to "1" to loop-back transmit channel 0 to receiver 1
0	ALOOP0	R/W	0	This bit is set to "1" to loop-back transmit channel 0 to receiver 0

## ARINC 429 Bit ordering

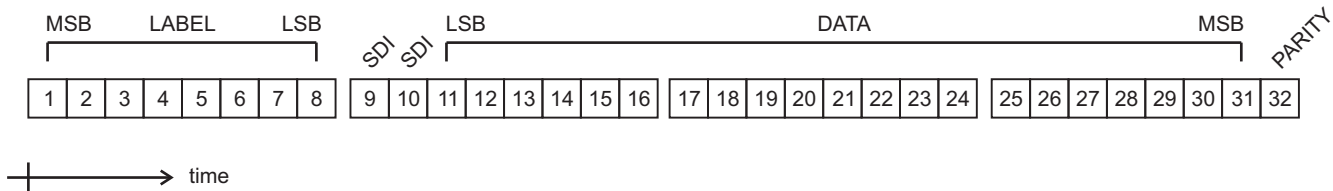
ARINC 429 messages consist of a 32-bit sequence as shown below. The first eight bits that appear on the ARINC 429 bus are the label byte. The next twenty three bits comprise a data field which presents data in a variety of formats defined in the ARINC 429 specification. The last bit transmitted is an odd parity bit.

The HI-3210 stores the received message as four bytes. The bytes are stored in memory in little-endian order. That is to say, the label byte (or status byte) is stored at the lowest memory address, the byte representing received bits 9 through 16 is stored at the next address, the byte representing bits 17 through 24 at the next address and the byte representing bits 25 through 32 at the highest address.

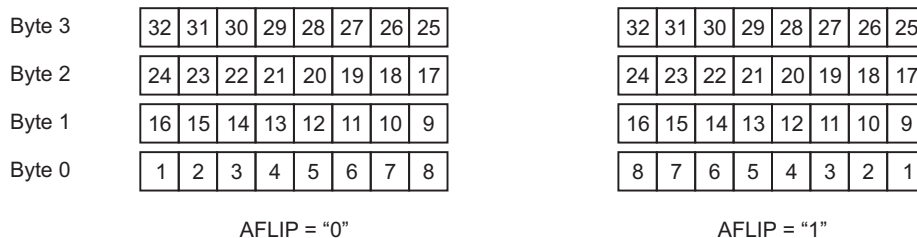
The ARINC 429 specifies the MSB of the label as ARINC bit 1. Conversely, the data field MSB is bit 31. So the bit significance of the label byte and data fields are opposite.

The HI-3210 may be programmed to "flip" the bit ordering of the label byte as soon as it is received and immediately prior to transmission. This is accomplished by setting the AFLIP bit to a "1" in the Master Control Register. Note that once the label byte has been flipped, the HI-3210 handles the flipped data byte "post-flip" for the purpose of label interrupt matching, filtering and storage.

## ARINC 429 Message as received / transmitted on the ARINC 429 serial bus



## ARINC 429 Message as stored in HI-3210 memory

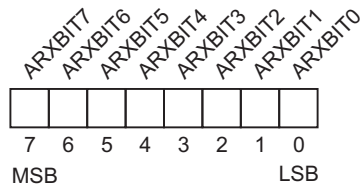


## ARINC 429 Bit Monitor Pins

The HI-3210 has the capability of externally monitoring any ARINC 429 received payload bit through the pins ARXBIT[7:0]. When the appropriate ARINC 429 receiver is enabled and the target label is received, the monitored bit value will be reflected on the pin. This allows the user to

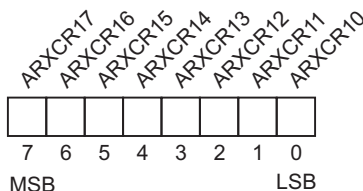
monitor any ARINC 429 received payload bit without performing any host SPI reads. The following registers configure the functionality of these monitor pins. Note that all these control register bits are RESET to zero.

### PINS ARXBIT[7:0] REGISTER (Address 0x805F)



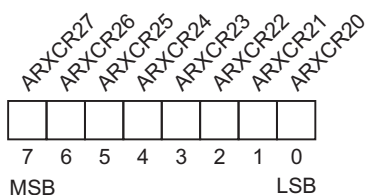
Bit	Name	R/W	Default	Description
7:0	ARXBIT[7:0]	R/W	0	These bits reflect the value of the corresponding pins ARXBIT[7:0]. After reset, all values are zero. When a monitored ARINC 429 bit changes, this register is updated with the value, which is reflected on the corresponding pin. The purpose of this register is to allow the user to preset the ARXBIT values after chip reset.

### PIN ARXBIT0 CONFIGURATION REGISTER 1 (Address 0x8060)



Bit	Name	R/W	Default	Description
7:5	ARXCR1[7:5]	R/W	0	These bits select which receive channel (0 through 7) will have bits monitored and reflected on the pin ARXBIT0.
4:0	ARXCR1[4:0]	R/W	0	These bits select which bit (8 through 31) of the ARINC payload will be reflected on the pin ARXBIT0. The receiver is specified by bits ARX0CR1[7:5] and the target label is specified by pin ARXBIT0 Configuration Register 2 described below. Note that bits 0 through 7 of the ARINC payload are not monitored and selecting these bits results in no effect.

### PIN ARXBIT0 CONFIGURATION REGISTER 2 (Address 0x8061)



Bit	Name	R/W	Default	Description
7:0	ARXCR2[7:0]	R/W	0	These bits select which label (0 through 255) will have bits monitored and reflected on the pin ARXBIT0. The receive channel and specific bits monitored are specified in ARXBIT0 Configuration Register 1 described above.

### PINS ARXBIT1 Through ARXBIT7 CONFIGURATION REGISTERS (Addresses 0x8062 to 0x806F)

Each pin ARXBIT1 through ARXBIT7 are also specified by a pair of configuration registers similar to ARXBIT0 described above. Functionality is exactly the same. The register addresses for each pin specification are listed in the Register Map section (see page 9). Note that HI-3210 provides external monitoring of eight bits through pins ARXBIT7 to ARXBIT0.

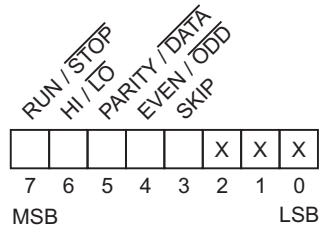
## ARINC 429 TRANSMIT OPERATION

The HI-3210 has four on-board ARINC 429 transmit channels which directly drive ARINC 429 differential line drivers such as the Holt HI-8596. ARINC 429 words may be written to the transmitters either directly, using an SPI instruction, or be generated automatically using the four ARINC 429 message schedulers.

### ARINC 429 Transmit Channel Configuration

Each of the four available ARINC 429 Transmit channels is configured using its own register. Register address 0x8018 controls ARINC 429 Transmit channel #0, register address 0x8019 controls channel #1 and so on. The ATXCn registers may be written or read at any time.

**ARINC 429 TX CONTROL REGISTER 0 - 3**  
(Address 0x8018 - 0x801B)



Bit	Name	R/W	Default	Description
7	RUN/ $\overline{\text{STOP}}$	R/W	0	When zero, transmission from this ARINC 429 transmit channel is suspended after the currently transmitting label is sent. When this bit is taken high, transmission starts at the beginning of the descriptor table for this channel.
6	$\overline{\text{HI}}$ /LO	R/W	0	Selects the transmission rate for the ARINC 429 transmit channel. A "0" selects high-speed (100Kb/s) and a "1" selects low-speed (12.5Kb/s).
5	PARITY/ $\overline{\text{DATA}}$	R/W	0	When this bit is a one, the 32nd transmitted ARINC bit is overwritten with a parity flag. When this bit is a zero, all 32-bits are transmitted as data.
4	EVEN/ $\overline{\text{ODD}}$	R/W	0	When PARITY / DATA is a "1", this bit defines whether th 32nd transmitted bit is set for Even or Odd Parity. A "1" selects even parity and a "0" selects odd parity.
3	SKIP	R/W	0	When set a "1" instructs the transmit sequencer to wait for the next Repetition Rate Counter rollover before beginning a new transmission cycle. A "0" causes an immediate restart of the cycle following completion of the prior cycle.
2	-	R/W	0	Not Used
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

## ARINC 429 Transmit Scheduler

Each of the four ARINC 429 transmit channels has its own transmit controller. The controller is user-programmed to output ARINC labels in a predefined order and repetition rate. A sequence of up to 256 ARINC labels may be transmitted before repeating the sequence.

A descriptor table with up to 256 entries (descriptors) is compiled by the user to define the sequence of ARINC 429 messages transmitted on each channel. When the RUN/STOP bit in the ARINC TX Control Register is asserted, the controller compiles the first 32-bit ARINC word from the instructions given by the first descriptor and then transmits it. A Transmit Sequence Pointer then increments to the next descriptor in the table and the process is repeated for Descriptor number 2.

ARINC 429 messages continue to be compiled and transmitted until the last descriptor in the table. The end of the table is marked by a special descriptor if not all 256 entries are needed. The Sequence Pointer is then reset to zero.

A Repetition Rate Counter is used to time the start of the next transmission cycle.

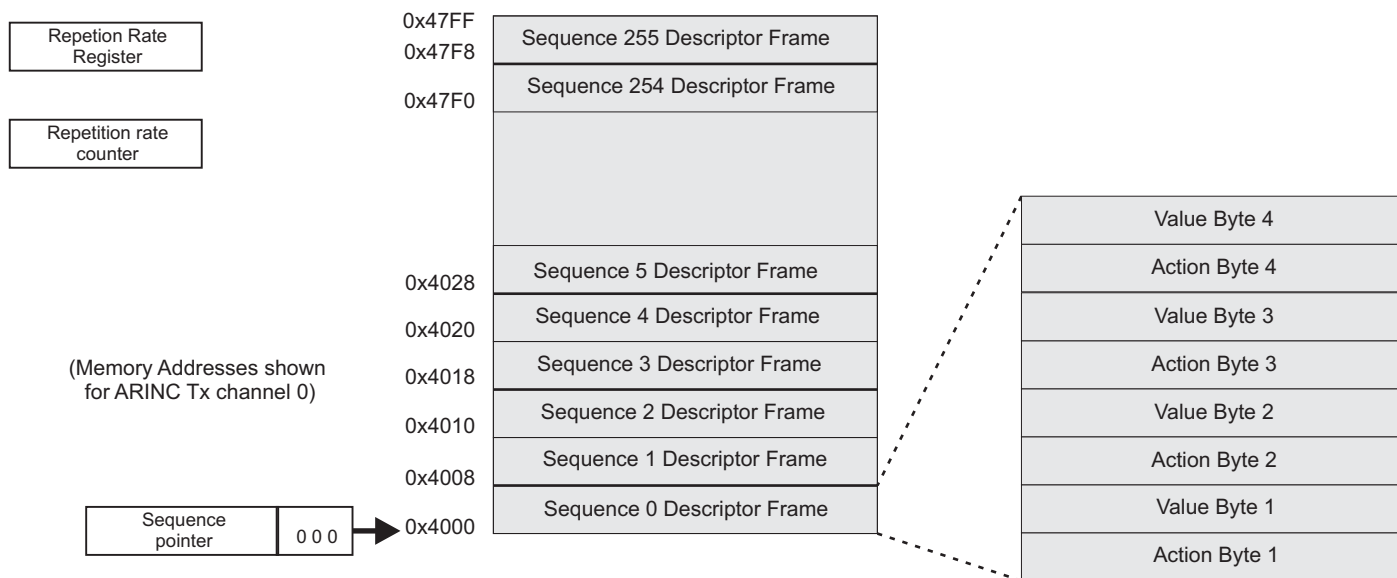
The user is responsible for construction of the descriptor table and for setting the Repetition Rate prior to asserting RUN/STOP. Facilities exist for immediate cycle repetition and for single-cycle operation.

The byte content of each ARINC 429 message transmitted is user defined by the descriptor contents. Data bytes may be sourced from the host CPU / auto-initialization EEPROM (immediate data) or from the ARINC 429 receive memory (ARINC indexed). This allows received ARINC data to be re-transmitted on another bus with or without filtering, label byte re-assignment or data modification. It also allows data from multiple ARINC 429 receive buses to be re-packetized into new ARINC 429 transmitted messages.

Conditional transmission control allows sequenced words to be skipped if no new data is available.

Each ARINC 429 transmit channel is independently configured with its own ARINC 429 TX Control Register, ATXCR0-3, as previously described.

## ARINC 429 Transmit Descriptor table



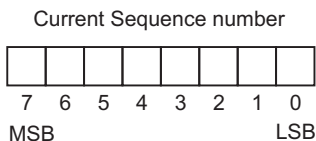
The value of each ARINC 429 label transmitted in the sequence is defined by its eight-byte descriptor. The descriptor consists of one "Action byte" and one "Value" byte for each of the four bytes that make up the ARINC 429 transmitted label.

The four pairs of Action and Value bytes describe where the data for each byte may be found. Different op-codes allow the data source to be host CPU populated fixed

values, or values from specific locations within the ARINC 429 receive memory. Action byte 1 also has one additional op-code to facilitate sequence flow control.

The construction of Action and Value bytes are described in the next section.

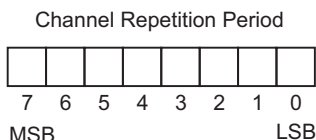
## TRANSMIT SEQUENCE POINTERS 0 -3 (Address 0x802C - 0x802F)



The transmit sequence pointer is set to zero on Master Reset. Once the Control Register RUN /  $\overline{\text{STOP}}$  bit goes high, sequence execution begins at sequence count zero. After the first word is sent, the pointer is incremented by one descriptor (counts descriptor frames).

This continues until the programmed sequence is complete. The sequence pointer is then reset to the beginning of the descriptor table and program execution begins as soon the channel repetition rate counter time elapses.

## REPETITION RATE REGISTER (Address 0x801C - 0x801F)



The Repetition rate register value defines the time interval between successive starts of the programmed transmit sequence for each ARINC 429 transmit channel. The value is set in binary, with the LSB representing 10 ms. Repetition rate time periods may therefore be set from 0 ms to 2.55 seconds

If the repetition rate is shorter than the minimum time needed to transmit all ARINC 429 words in the sequence (but not zero), the transmit sequence will begin again immediately if the Control Register SKIP bit is a zero. If the SKIP bit is a one, the sequencer will wait until the next rollover of the Repetition Rate Counter before starting a new cycle.

## One-time Sequence Transmission

When the Repetition Rate Register contains zero (default), sequence transmission occurs just once, upon 0-to-1 transition of the RUN/STOP bit in the Transmit Control Register. One-time execution of the sequencer is useful when transmitting ARINC 429 words directly from the host CPU. One or more immediate-mode descriptors can be written into the sequence table and transmitted, then transmit values can be refreshed for the next cycle.

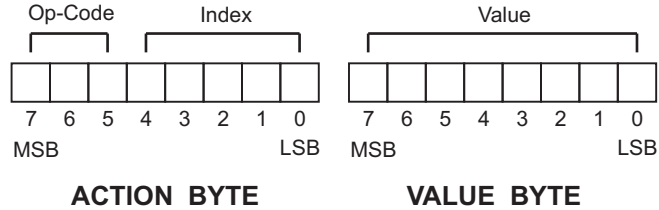
There are three ways to retrigger one-time transmission after its previous completion:

**Method 1:** Reset the RUN/STOP bit in the Transmit Control Register to zero. Sequentially write 0xFFFF then 0x0000 into the Repetition Rate Register. Initiate one-time transmission by setting the RUN/STOP bit in the Transmit Control Register; retransmission occurs immediately. This method can be used without affecting the other three transmitters.

**Method 2:** With the RUN/STOP bit in the Transmit Control Register already set to logic-1 by the previous one-time sequence transmission, toggle the state of the RUN input pin low and then high; retransmission occurs immediately.

**Method 3:** With the RUN/STOP bit in the Transmit Control Register already set to logic-1 by the previous one-time sequence transmission, toggle the state of A429TX bit 6 in the Master Control Register low and then high; retransmission occurs immediately.

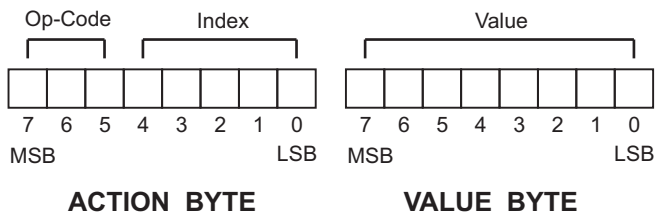
## ARINC 429 BYTE 1 DESCRIPTOR



<u>Op-Code</u>	<u>Index</u>	<u>Value</u>	<u>Description</u>
000	XXXXX	XXXXXXXX	End of sequence. When op-code 000 is encountered by the sequencer before it reaches sequence number 255, the sequencer resets to zero and begins the next transmission cycle starting at descriptor number 0 as soon as the repetition rate counter rolls over. Note that the descriptor table is cleared following Master Reset, so no ARINC 429 transmissions are possible until the sequence table has been configured.
001	XXXXX	XXXXXXXX	No-operation. This descriptor is ignored and the sequencer increments to the next descriptor in the sequence. This opcode may be used to temporarily suspend transmission of a particular message in the sequence, without having to modify the remaining bit fields of the descriptor or reloading the entire descriptor block in order to delete the entry.
010	XXXXX	LLLLLLLL	Immediate data. The value contained in the descriptor value data byte is loaded into byte 1 (the ARINC 429 "label" byte) of the ARINC 429 label to be transmitted.
011	CCCXX	LLLLLLLL	Immediate data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel "CCC" and label block "LLLLLLLL" is read. LLLLLLLL is used as Byte 1 if NEWTXn is set for this <b>or any other conditional opcode</b> within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
100	CCCB	LLLLLLLL	Indexed data. The value of ARINC Data RAM location defined by channel "CCC", label block "LLLLLLLL" and byte number "BB" is loaded into byte 1 of the ARINC 429 label to be transmitted.
101	CCCB	LLLLLLLL	Indexed data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel "CCC", label block "LLLLLLLL" and byte number "BB" is read. The corresponding byte is used as Byte 1 if NEWTXn is set for this <b>or any other conditional opcode</b> within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
110	XXXX	XXXXXXXX	Reserved. Do not use.
111	XXXX	XXXXXXXX	Reserved. Do not use.



ARINC 429 BYTES 2 - 4 DESCRIPTOR



<u>Op-Code</u>	<u>Index</u>	<u>Value</u>	<u>Description</u>
000	XXXXX	XXXXXXXXX	No-Op op-code. ARINC 429 word construction will be terminated and the sequencer will move on to the next descriptor in the table.
001	XXXXX	XXXXXXXXX	No-Op op-code. ARINC 429 word construction will be terminated and the sequencer will move on to the next descriptor in the table.
010	XXXXX	LLLLLLLLL	Immediate data. The value contained in the descriptor value data byte is loaded into this byte position of the ARINC 429 32-bit message to be transmitted.
011	CCCXX	LLLLLLLLL	Immediate data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC” and label block “LLLLLLLLL” is read. LLLLLLLLL is used if NEWTXn is set for this <b>or any other conditional opcode</b> within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
100	CCCB	LLLLLLLLL	Indexed data. The value of ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLLL” and byte number “BB” is loaded into this byte position of the ARINC 429 label to be transmitted.
101	CCCB	LLLLLLLLL	Indexed data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLLL” and byte number “BB” is read. The corresponding byte is used if NEWTXn is set for this <b>or any other conditional opcode</b> within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
110	XXXX	XXXXXXXXX	Reserved. Do not use.
111	XXXX	XXXXXXXXX	Reserved. Do not use.



## ARINC 429 Immediate Transmit Option

The Host CPU may instruct the HI-3210 to transmit an ARINC 429 message immediately using a special SPI command. The SPI command selects the transmit channel and provides the four bytes of data to be sent as a 32-bit ARINC 429 message.

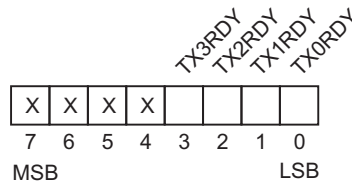
If the transmit channel's sequencer is not running (ATCR bit RUN/STOP = "0"), or the sequencer is waiting for the repetition rate counter to rollover, then the new ARINC 429 message is transmitted without delay.

If the transmit sequencer for the selected channel is active, then the new message is transmitted as soon as the current message has been sent. The sequencer then resumes operation at the next location in the queue.

Both the RUN input and the Master Control Register A429TX bit must be high to enable any ARINC 429 transmission.

Table 1 lists the host CPU SPI instruction format.

### ARINC 429 TX READY BITS REGISTER (Address 0x800D)



Bit	Name	R/W	Default	Description
7	-	R	0	Not Used
6	-	R	0	Not Used
5	-	R	0	Not Used
4	-	R	0	Not Used
3	TX3RDY	R	1	This bit is set when Transmit Buffer 3 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 3 is not empty.
2	TX2RDY	R	1	This bit is set when Transmit Buffer 2 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 2 is not empty.
1	TX1RDY	R	1	This bit is set when Transmit Buffer 1 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 1 is not empty.
0	TX0RDY	R	1	This bit is set when Transmit Buffer 0 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 0 is not empty.

## RESET AND START-UP OPERATION

After power-on, the HI-3210 is in an undefined state. The RESET pin must be taken high to begin device initialization. The RESET pin may be asserted at any time. Taking RESET high immediately stops all execution and sets the READY output low indicating that the part is in the reset state.

On the falling edge of RESET, the HI-3210 samples the state of the MODE2-0 input pins. This is the only occasion these inputs are sampled. The state of the MODE pins determines one of six possible initialization sequences as shown in the following diagram. These six initialization modes allow the user to customize the start-up configuration of the device. **For Modes 2 and 3, the RBFFAIL bit-7 should be cleared to 0 by writing to the BIST Control/Status register (Address 0x8070) BEFORE taking RESET high.**

Once the initialization is complete, the device enters the Idle State when the ready pin goes high. In Idle State, the host CPU may communicate with the HI-3210 memory and registers using the host CPU SPI link. When in the Idle State, the HI-3210 does not transmit or receive any messages on any of the ARINC 429 buses.

To begin data bus operation, the user must transition the RUN input from a low to high state. Immediately following the rising edge of RUN, the part enters the Active State and bus message processing begins. During initialization, various device configuration tasks are performed according to the Mode selection set at the MODE2:0 input pins. The available options are:

### 1. RAM Integrity Check

In Modes 2 and 3, the HI-3210 performs a RAM integrity check. A read/write check is performed on the entire RAM space. An incrementing pattern is written to sequential RAM locations then this pattern is read and verified. Each RAM location is re-written with the 1s complement of its current contents then this pattern is read and verified. The incrementing pattern followed by its 1s complement ensures that each RAM location can store both a 1 and 0 state. If the RAM integrity check fails, the MINT pin is asserted and the Pending Interrupt Register RAMFAIL bit is set. The part enters the "Safe" state, in which the HI-3210 is able to accept and respond to Host CPU SPI Instructions, but cannot enter Normal Operating mode until the RESET input is taken high to repeat the initialization sequence. The RAMFAIL Interrupt is not maskable.

### 2. Clear Data Memory

In Modes 0, 1, 2, 3 and 5, the HI-3210 automatically clears all memory locations in the address range 0x0000 to 0x33FF. This is the space reserved for ARINC 429

message data. Configuration tables and HI-3210 registers are not affected.

### 3. Initialize Registers and Clear all memory

In addition to clearing data memory (0x0000 to 0x33FF), Modes 0, 1, 2, and 3 also clear all configuration and look-up tables (0x3400 to 0x7FFF) as well as setting all registers (0x8000 to 0x807F) to their default states. All registers default to zero unless otherwise noted.

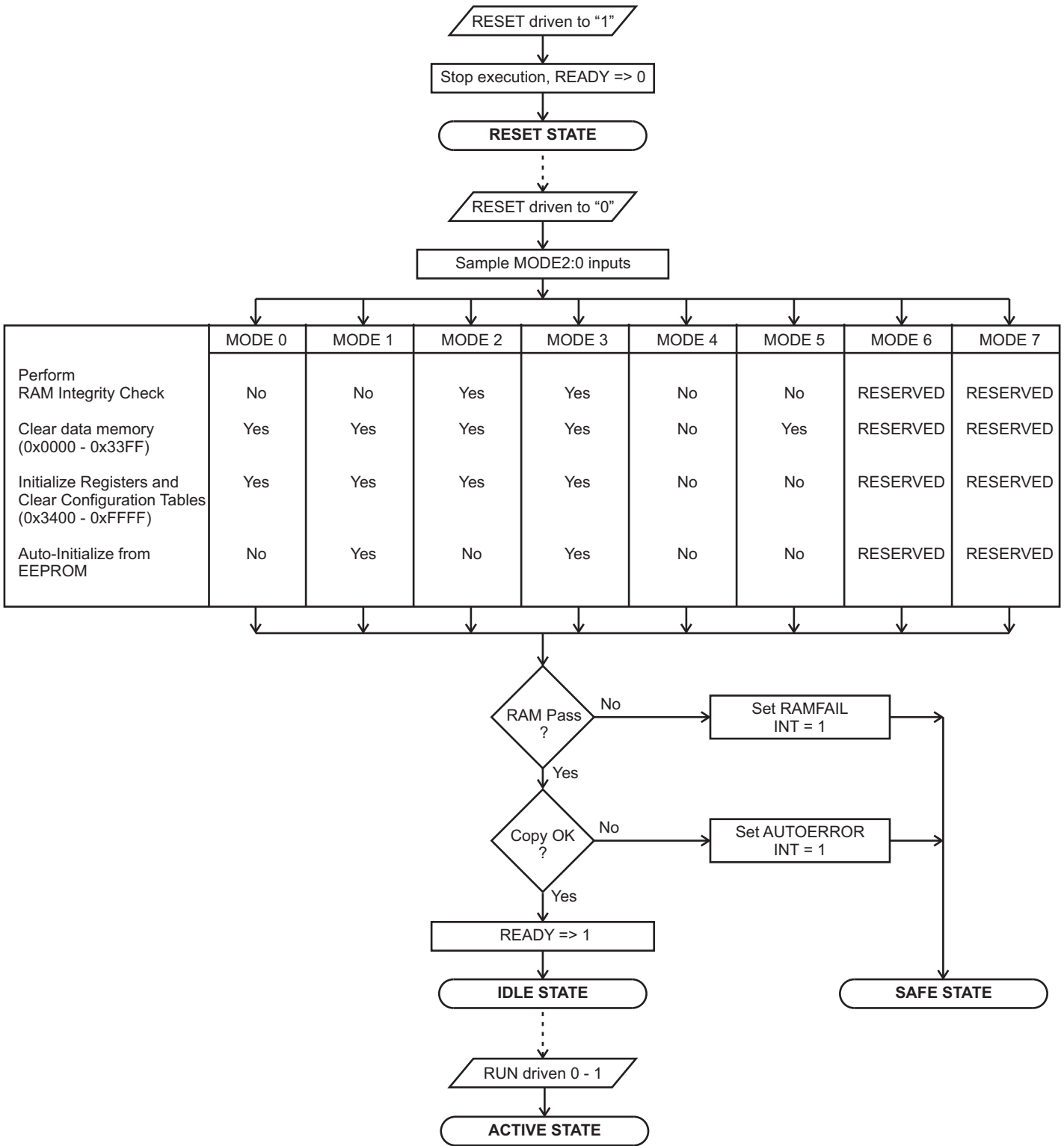
### 4. Auto-Initialize from EEPROM

The contents of the Auto-Initialization EEPROM are copied into the HI-3210 memory and registers via the EEPROM SPI interface. The part verifies the integrity of the data transfer from the EEPROM by running through a byte-by-byte compare routine and a checksum validation. If a compare error is detected, the AUTOERR bit is set in the Pending Interrupt Register, the MINT output is asserted, the location of the error is captured in the AUTO-INIT FAIL ADDRESS registers 0x8073 (Auto-Init Fail LS address) and 0x8074 (Auto-Init Fail MS address) and the part enters the Safe state. If a checksum error is detected, the CHKERR bit is set in the Pending Interrupt Register, the MINT output is asserted and the part enters the Safe state. The AUTOERR and the CHKERR interrupts are not maskable.

Once initialization is complete, the part enters the Idle state. The host CPU may read and write HI-3210 internal memory and registers in all Modes. If not using the auto-initialization feature, the host CPU should configure the device at this time.

**NOTE:** Modes 6 and 7 are reserved and should not be used.

## Reset and Start-up Operation



## INTERRUPT HANDLING

The HI-3210 includes a simple, user-selectable Interrupt Handler. Two types of Interrupt are possible - Message Event Driven (ARINC 429 Bus), and Fault Driven.

### ARINC 429 Receive Interrupts

As described earlier, the user can elect to generate an interrupt upon receipt of an ARINC 429 message on any combination of the eight available channels and for any of the possible 256 label byte (ARINC message bits 1-8) values. Interrupts are enabled when the ARINC 429 Rx Interrupt look-up bit is a "1".

When a message arrives that is flagged to generate an Interrupt, that channel's bit is set in the ARINC 429 Receiver Pending Interrupt Register APIR. The ARINC 429 Interrupt Address Register (AIAR) for that channel is updated with the ARINC 429 8-bit label value.

For example, if ARINC Receive channel 7 is enabled for Interrupts when messages with ARINC label 0xD4 arrive, then on receipt of such a message, APIR bit 7 is set to a "1" and the value 0xD4 is written to AIAR7.

If the corresponding bit in the ARINC 429 Receive Interrupt Mask Register is a "1" the AINT interrupt output will go high and stay high until the AACK input pin is driven high. Driving AACK high, causes the AINT pin to return to zero.

A special Indexed SPI read instruction is available to allow the host to efficiently retrieve ARINC 429 messages which have Interrupts Enabled (see SPI instruction set section).

Note that if AACK is tied high permanently, the AINT pin will go high for approximately 1 us before returning to zero. A host CPU read of the APIR register reads the current value and resets APIR to 0x00.

### Fault Interrupts

There are four fault Interrupt bits in the PIR. Fault Interrupts are not maskable, and their Interrupt Mask bits are fixed at a "1".

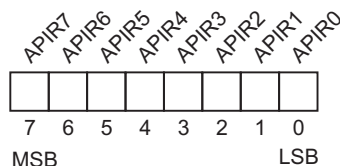
COPYERR is set when the HI-3210 detects a mismatch between RAM and EEPROM after attempting to program the Auto-initialization EEPROM.

AUTOERR is set when the Auto-Initialization EEPROM read verification cycle detects a mismatch between the on-chip memory and EEPROM following auto-initialization.

CHKERR is set when an auto-initialization checksum error is detected.

The RAMFAIL bit is set if the Built-In Self Test sequence fails.

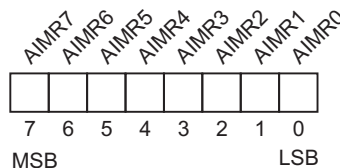
**ARINC 429 RECEIVE PENDING INTERRUPT REGISTER**  
(Address 0x8000)



Bit	Name	R/W	Default	Description
7:0	APIR[7:0]	R	0	When a message is received on a given channel that triggers an interrupt, that channel's corresponding bit is set, e.g. if a message received on Rx channel 5 triggers an interrupt, the bit APIR5 will be set. If this bit is unmasked in the ARINC 429 Receive Interrupt Mask Register (see below), the AINT output pin is asserted.

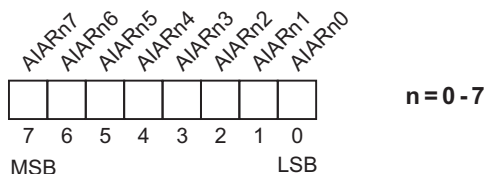
**Note:** This register is automatically cleared when read.

**ARINC 429 RECEIVE INTERRUPT MASK REGISTER**  
(Address 0x8020)



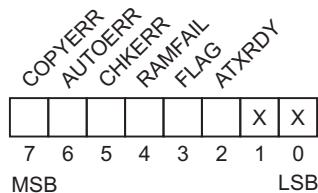
Bit	Name	R/W	Default	Description
7:0	AIMR[7:0]	R/W	0	Each bit in this register, AIMR[7:0], is a mask for a corresponding bit in the ARINC 429 Receive Pending Interrupt Register (APIR[7:0], described above). Writing a "1" to an AIMR bit results in assertion of the AINT output pin when the corresponding APIR bit is set (ARINC 429 message received). Writing a "0" to an AIMR bit will mask the corresponding APIR bit in the ARINC 429 Receive Pending Interrupt Register, resulting in non-assertion of the AINT pin when an ARINC 429 message is received.

**ARINC 429 RECEIVE INTERRUPT ADDRESS REGISTERS 0 - 7**  
(Address 0x8001 - 0x8008)



Bit	Name	R/W	Default	Description
7:0	AIAR0[7:0]	R	0	The label number of the ARINC 429 message causing an interrupt is loaded into this register. Each channel has a corresponding ARINC 429 Receive Interrupt Address Register (AIAR0 - AIAR7).

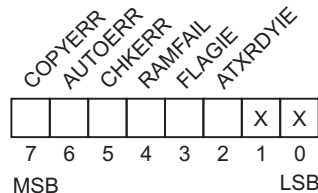
**PENDING INTERRUPT REGISTER**  
(Address 0x800A)



The MINT will be asserted when any of the bits in this register are set.

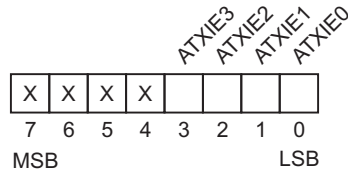
Bit	Name	R/W	Default	Description
7	COPYERR	R	0	EE copy error. RAM - EEPROM mismatch
6	AUTOERR	R	0	Auto-initialization RAM read error
5	CHKERR	R	0	Auto-initialization checksum fail
4	RAMFAIL	R	0	Power-On Reset RAM Integrity Check fail
3	FLAG	R	0	Logical OR of ARINC 429 Receive FIFO FLAG signals
2	ATXRDY	R	0	ARINC 429 Host TX ready. Used with Host SPI op-code 100101TT (see Table 1). Interrupt when any of the four ARINC 429 transmitters are ready for the next 32-bit word from the host
1	-	R	0	Not Used
0	-	R	0	Not Used

**PENDING INTERRUPT ENABLE REGISTER**  
(Address 0x8034)



Bit	Name	R/W	Default	Description
7	COPYERR	R	1	COPYERR is not maskable
6	AUTOERR	R	1	AUTOERR is not maskable
5	CHKERR	R	1	CHKERR is not maskable
4	RAMFAIL	R	1	RAMFAIL is not maskable
3	FLAGIE	R/W	0	MINT pin is asserted if this bit is a "1" and the Pending Interrupt Register FLAG bit is set
2	ATXRDYIE	R/W	0	MINT pin is asserted if this bit is a "1" and the Pending Interrupt Register ATXRDY bit is set
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

**ARINC 429 TX READY INT ENABLE**  
**(Address 0x8035)**

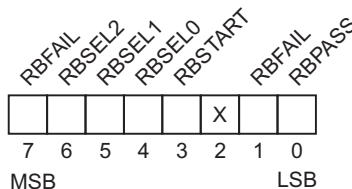


Bit	Name	R/W	Default	Description
7	-	R/W	0	Not Used
6	-	R/W	0	Not Used
5	-	R/W	0	Not Used
4	-	R/W	0	Not Used
3	ATXIE3	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 3 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
2	ATXIE2	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 2 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
1	ATXIE1	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 1 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
0	ATXIE0	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 0 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.

## RAM BUILT-IN SELF-TEST

The HI-3210 offers a built-in self-test (BIST) feature which can be used to check RAM integrity. The BIST Control/Status Register is used to control the BIST function. All tests are destructive, overwriting data present before test commencement.

### BIST CONTROL/STATUS REGISTER (Address 0x8070)



This register controls RAM built-in self-test. Bits 0,1 are Read Only. The remaining bits in this register are Read-Write but can be written only in MODE2:0 = 0x04.

BIST Control Register bits provide a means for the host to perform RAM self-test at other times. Register bits 6:4 select RAM test type. Then bit 3 starts the selected RAM test, and bits 1:0 report a fail/pass result after test completion.

*Bit No. Mnemonic Interrupt Type*

- |     |          |   |
|-----|----------|---|
| 7   | RBFFAIL  | RAM BIST Force Failure.<br>When this bit is asserted, RAM test failure is forced to verify that RAM BIST logic is functional.<br><b>For Modes 2 and 3, the RBFFAIL bit-7 should be cleared to 0 by writing to the BIST Control/Status register (Address 0x8070) BEFORE taking RESET high.</b> |
| 6:4 | RBSEL2-0 | RAM BIST Select Bits 2-0.<br>This 3-bit field selects the RAM BIST test mode applied when the RBSTART bit is set:   |

#### RBSEL2:0 SELECTED RAM TEST

000	Idle
001	Pattern Test, described below
010	Write 0x00 to RAM address range 0x0000 - 0x7FFF
011	Read and verify 0x00 over RAM address range 0x0000 - 0x7FFF
100	Write 0xFF to RAM address range 0x0000 - 0x7FFF
101	Read and verify 0xFF over RAM address range 0x0000 - 0x7FFF
110	Inc / Dec Test performs only steps 5 - 8 of the Pattern Test below
111	Idle

#### Description of the RAM BIST "PATTERN" test selected when register bits RBSEL2:0 = 001:

1. Write 0x00 to all RAM locations, 0x0000 through 0x7FFF
2. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:
  - a. Read and verify 0x00
  - b. Write then read and verify 0x55
  - c. Write then read and verify 0xAA
  - d. Write then read and verify 0x33
  - e. Write then read and verify 0xCC
  - f. Write then read and verify 0x0F
  - g. Write then read and verify 0xF0
  - h. Write then read and verify 0x00
  - i. Write then read and verify 0xFF
  - j. Write 0x00 then increment RAM address and go to step (a)
3. Write 0xFF to all RAM locations, 0x0000 through 0x7FFF
4. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:
  - a. Read and verify 0xFF
  - b. Write then read and verify 0x55
  - c. Write then read and verify 0xAA



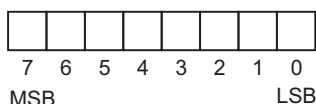
- d. Write then read and verify 0x33
- e. Write then read and verify 0xCC
- f. Write then read and verify 0x0F
- g. Write then read and verify 0xF0
- h. Write then read and verify 0x00
- i. Write then read and verify 0xFF
- j. Write 0xFF then increment RAM address and go to step (a)

- 5. Write an incrementing pattern into sequential RAM locations from 0x0000 to 0x7FFF
- 6. Read each memory location from 0x0000 to 0x7FFF and verify the contents
- 7. Write 1s complement of each cell's current contents, into each RAM location (same addr range)
- 8. Read each memory location and verify the contents

3	RBSTRT	<p>RAM BIST Start.</p> <p>Writing logic 1 to this bit initiates the RAM BIST test selected by register bits RBSEL2:0. The RBSTRT bit can only be set in MODE2:0 = 0x04. This bit is automatically cleared upon test completion. Register bits 1:0 indicate fail / pass test result.</p>
2	-----	Not Used.
1	RBFAIL	<p>RAM BIST Fail.</p> <p>Device logic asserts this bit when failure occurs while performing the selected RAM test. This bit is automatically cleared when RBSTRT bit 3 is set. When BIST failure occurs, a clue to the failing RAM address can be read at register addresses 0x8071 and 0x8072. For speed, the RAM BIST concurrently tests four consecutive RAM addresses in parallel. If a test failure occurs, register addresses 0x8071 and 0x8072 can be used to determine the four RAM addresses tested.</p>
0	RBPASS	<p>RAM BIST Pass.</p> <p>Device logic asserts this bit when the selected RAM test completes without error. This bit is automatically cleared when RBSTRT bit 3 is set.</p>

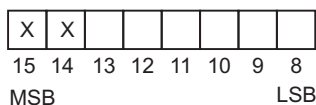
**BISTFL**

**LOWER BIST FAIL ADDRESS REGISTER**  
(Address 0x8071)



**BISTFH**

**UPPER BIST FAIL ADDRESS REGISTER**  
(Address 0x8072)



## HOST SERIAL PERIPHERAL INTERFACE

In the HI-3210, internal RAM and registers occupy a (32K + 128) x 8 address space. The lowest 32K addresses access RAM locations and the remaining addresses access registers. Timing is identical for register operations and RAM operations via the serial peripheral interface, and read and write operations have likewise identical timing.

Host access is only allowed when the part is READY or in SAFE mode. **NOTE:** writes will be blocked and reads will return the Master Status Register value until either of these modes occur.

### Serial Peripheral Interface (SPI) Basics

The HI-3210 uses an SPI synchronous serial interface for host access to registers and RAM. Host serial communication is enabled through the Chip Select ( $\overline{CS}$ ) pin, and is accessed via a three-wire interface consisting of Serial Data Input (SI) from the host, Serial Data Output (SO) to the host and Serial Clock (SCK). All programming cycles are completely self-timed, and no erase cycle is required before write.

The SPI (Serial Peripheral Interface) protocol specifies master and slave operation; the HI-3210 Host CPU interface operates as an SPI slave.

The SPI protocol defines two parameters, CPOL (clock polarity) and CPHA (clock phase). The possible CPOL-CPHA combinations define four possible "SPI Modes." Without describing details of the SPI modes, the HI-3210 operates in the two modes where input data for each device (master and slave) is clocked on the rising edge of

SCK, and output data for each device changes on the falling edge. These are known as SPI Mode 0 (CPHA = 0, CPOL = 0) and SPI Mode 3 (CPHA = 1, CPOL = 1). Be sure to set the host SPI logic for one of these modes.

As seen in Figure 1, the difference between SPI Modes 0 and 3 is the idle state for the SCK signal. There is no configuration setting in the HI-3210 to select SPI Mode 0 or Mode 3 because compatibility is automatic. Beyond this point, the HI-3210 data sheet only shows the SPI Mode 0 SCK signal in timing diagrams.

The SPI protocol transfers serial data as 8-bit bytes. Once  $\overline{CS}$  chip select is asserted, the next 8 rising edges on SCK latch input data into the master and slave devices, starting with each byte's most-significant bit. The HI-3210 SPI can be clocked at 20 MHz.

Multiple bytes may be transferred when the host holds  $\overline{CS}$  low after the first byte transferred, and continues to clock SCK in multiples of 8 clocks. A rising edge on  $\overline{CS}$  chip select terminates the serial transfer and reinitializes the HI-3210 SPI for the next transfer. If  $\overline{CS}$  goes high before a full byte is clocked by SCK, the incomplete byte clocked into the device SI pin is discarded.

In the general case, both master and slave simultaneously send and receive serial data (full duplex) as shown in Figure 1 below. When the HI-3210 is sending data on SO during read operations, activity on its SI input is ignored. Figures 2 and 3 show actual behavior for the HI-3210 SO output.

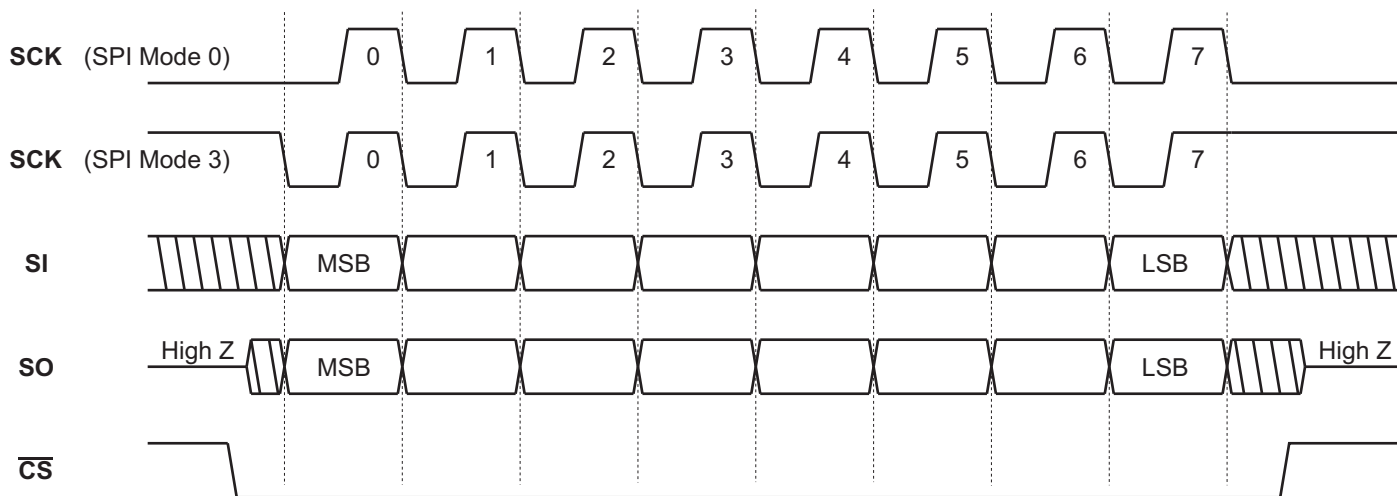


FIGURE 1. Generalized Single-Byte Transfer Using SPI Protocol, SCK is Shown for SPI Modes 0 and 3

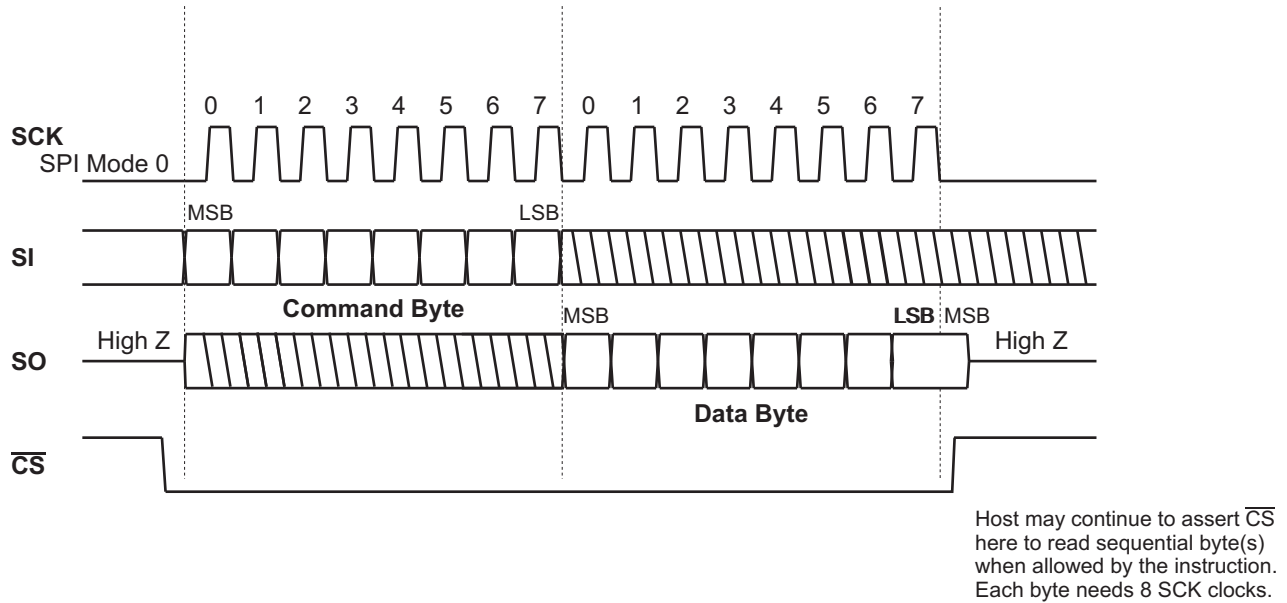


FIGURE 2. Single-Byte Read From RAM or a Register

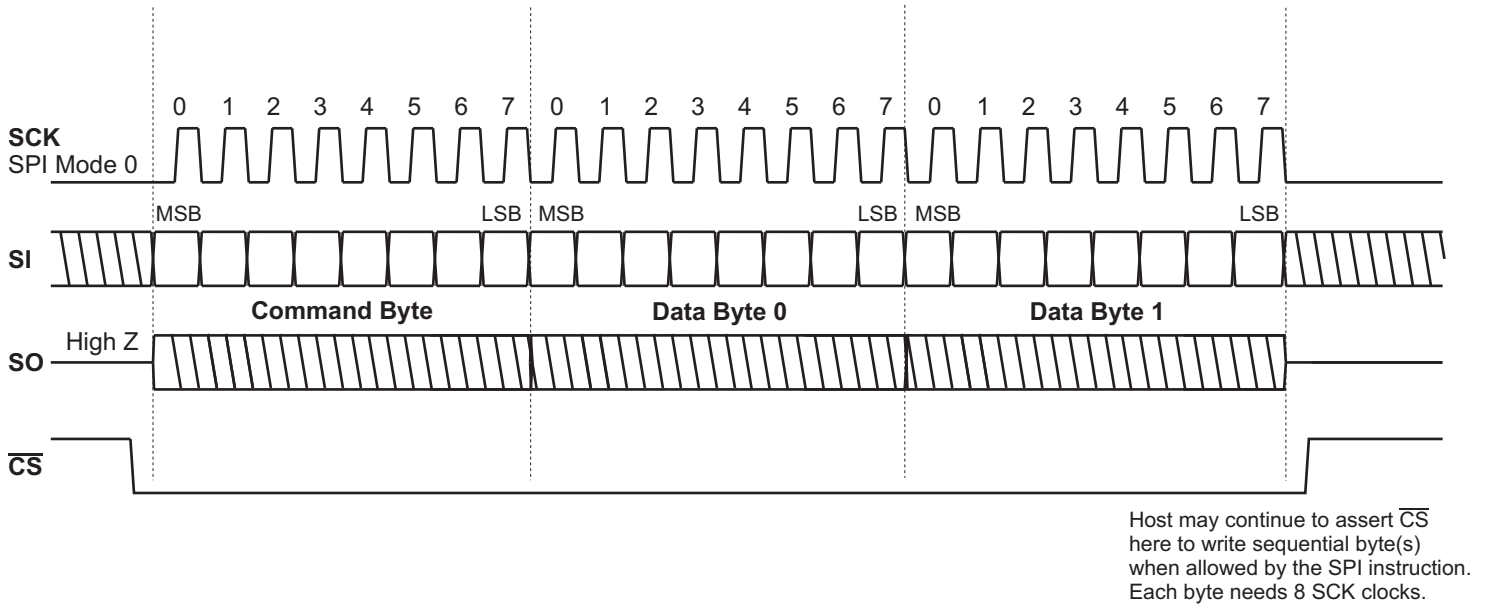


FIGURE 3. 2-Byte Write To RAM or a Register Pair

## HI-3210 SPI COMMANDS

For the HI-3210, each SPI read or write operation begins with an 8-bit command byte transferred from the host to the device after assertion of  $\overline{CS}$ . Since HI-3210 command byte reception is half-duplex, the host discards the dummy byte it receives while serially transmitting the command byte.

### Fast Access Commands for Registers 0-15

The SPI command set includes directly-addressed read and write commands for registers 0 through 15 (Memory Address 0x8000 to 0x800F). The 8-bit pattern for these commands has the general form

0-W-R-R-R-0-0

where RRRR is the 4-bit register number, and W signifies Write when 1, or Read when 0.

Figures 2 and 3 show read and write timing as it appears for fast-access register operations. The command byte is immediately followed by a data byte comprising the 8-bit data word read or written. For a single register read or write,  $\overline{CS}$  is negated after the data byte is transferred.

Multiple register read or write cycles may be performed by transferring more than one byte before  $\overline{CS}$  is negated. Multiple register access occur in address order starting with the register specified in the SPI instruction.

Note: Register locations not shown in table 1 are “reserved” and cannot be written using any SPI command. Further, these register addresses will not provide meaningful data in response to read commands.

### RAM and Register Indirect Addressing

Refer to the HI-3210 SPI command set shown in Table 1. SPI commands other than fast-access use an address pointer to indicate the address for read or write transactions. This sixteen-bit memory address pointer (MAP) must be initialized before any non-fast-access read or write operation. Two dedicated SPI instructions are used to write and read the MAP. SPI Instruction 0x8C followed by two data bytes is used to write MAP. SPI instruction 0x88 reads two data bytes from MAP. The first byte is the most significant eight bytes of the address. For example, SPI sequence 0x8C, 0x12, 0x34 write the value 0x1234 into the MAP.

Two SPI instructions read and write data bytes to memory or registers using the MAP as an address pointer. Single or multi-byte reads and writes may be performed. MAP is incremented after each byte access.

**Two command bytes cannot be “chained”;**  $\overline{CS}$  must be negated after the command, then reasserted for the following Read or Write command.

Note: When the primary or fast-access address pointer is used for auto-incrementing multi-word read/write and reaches the top of the memory address range (0x7FFF), or the top of the register address range (0xFFFF) attempts to read further bytes will result the terminal address (0x7FFF or 0xFFFF) being output again. The host should avoid this situation.

Two single-byte SPI commands use the current address pointer value in MAP without first loading or otherwise modifying it:

<b>Command</b>	<b>Read Operation</b>
0x80	read location addressed by pointer value

<b>Command</b>	<b>Write Operation</b>
0x84	write location addressed by pointer value

Either of these commands can be used to read or write a single location, or may be used when starting a multi-byte read or write by using the pointer’s auto-increment feature.

### Special Purpose Commands

Several other HI-3210 SPI commands load or otherwise modify the memory address pointer before initiating a read or write process. These commands are designed to allow speedy access to messages received on the ARINC 429 buses.

Using a single-byte SPI command, the address pointer can be directly loaded with the memory address for the last received ARINC 429 message which triggered an interrupt.

#### Op Code 110RRR00

The HI-3210 will retrieve the current ARINC Receive Interrupt Vector for a given channel (RRR), calculate the memory address for the first word of the corresponding receive memory data block and write it to the Memory Address pointer (MAP). Read the location addressed by the new pointer value.

This command can be used to read just the most recent ARINC 429 Receive Status Byte, or may be used to start a four-byte read because memory pointer auto-increment occurs after the Status Byte is read.

#### Op Code 111RRR00

The HI-3210 will retrieve the current ARINC Receive Interrupt Vector for a given channel (RRR), calculate the memory address for the first word of the corresponding receive memory data block and write it to the Memory Address Pointer (MAP). Output the value of the Receive Interrupt Vector (ARINC 429 label byte).

This command can be used to read just the most recent ARINC 429 label value received, or may be used to start a four-byte read to output the entire four-byte ARINC

message, because memory pointer auto-increment occurs after the label byte is output.

scheduler TT for immediate transmission, where TT represents the channel number.

### Op Code 100101TT

Writes an ARINC 429 message to ARINC 429 transmit

**TABLE 1. DEFINED INSTRUCTIONS**

OP CODE Binary	OP CODE Hex	Auto Increment	Number of Data Bytes	DESCRIPTION
00RRRR00	0x00 - 0x3C	Yes	1++	Fast Register Read from register RRRR
01RRRR00	0x40 - 0x7C	Yes	1++	Fast Register Write to register RRRR
10000000	0x80	Yes	1++	Read memory at address MAP
10000100	0x84	Yes	1++	Write memory at address MAP
10001000	0x88	No	2	Read MAP
10001100	0x8C	No	2	Write MAP
100101TT	0x94 - 0x97	No	4	Transmit ARINC 429 message on transmit bus TT
101RRR00	0xA0 - 0xBC	Yes	4, 8, 12...	Read ARINC 429 FIFO # RRR. Reads exactly four bytes
110RRR00	0xC0 - 0xDC	No	4	Read ARINC block at receive channel RRR, label <ARIn>
111RRR00	0xE0 - 0xFC	No	4	Read ARINC message at receive channel RRR, label <ARIn>

### FAST-ACCESS SPI COMMANDS FOR REGISTERS 0-15 Command Bits 5:2 Convey the 4-Bit Register Address

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FAST-ACCESS READ
0 0 0 0 0 0 0 0	0x00	Read APIR
0 0 0 0 0 1 0 0	0x04	Read AIAR0
0 0 0 0 1 0 0 0	0x08	Read AIAR1
0 0 0 0 1 1 0 0	0x0C	Read AIAR2
0 0 0 1 0 0 0 0	0x10	Read AIAR3
0 0 0 1 0 1 0 0	0x14	Read AIAR4
0 0 0 1 1 0 0 0	0x18	Read AIAR5
0 0 0 1 1 1 0 0	0x1C	Read AIAR6
0 0 1 0 0 0 0 0	0x20	Read AIAR7
0 0 1 0 0 1 0 0	0x24	Reserved
0 0 1 0 1 0 0 0	0x28	Read PIR
0 0 1 0 1 1 0 0	0x2C	Reserved
0 0 1 1 0 0 0 0	0x30	Read AMFF
0 0 1 1 0 1 0 0	0x34	Read ATRB
0 0 1 1 1 0 0 0	0x38	Read MSR
0 0 1 1 1 1 0 0	0x3C	Read MCR

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FAST-ACCESS WRITE
0 1 0 0 0 0 0 0	0x40	N/A (Read only)
0 1 0 0 0 1 0 0	0x44	N/A (Read only)
0 1 0 0 1 0 0 0	0x48	N/A (Read only)
0 1 0 0 1 1 0 0	0x4C	N/A (Read only)
0 1 0 1 0 0 0 0	0x50	N/A (Read only)
0 1 0 1 0 1 0 0	0x54	N/A (Read only)
0 1 0 1 1 0 0 0	0x58	N/A (Read only)
0 1 0 1 1 1 0 0	0x5C	N/A (Read only)
0 1 1 0 0 0 0 0	0x60	N/A (Read only)
0 1 1 0 0 1 0 0	0x64	Reserved
0 1 1 0 1 0 0 0	0x68	N/A (Read only)
0 1 1 0 1 1 0 0	0x6C	Reserved
0 1 1 1 0 0 0 0	0x70	N/A (Read only)
0 1 1 1 0 1 0 0	0x74	N/A (Read only)
0 1 1 1 1 0 0 0	0x78	N/A (Read only)
0 1 1 1 1 1 0 0	0x7C	Write MCR

## PROGRAMMING THE AUTO-INITIALIZATION EEPROM.

Following reset, the HI-3210 may be completely configured by automatically copying the contents of an external EEPROM into HI-3210 memory and registers. An SPI enabled 64KByte EEPROM is used for this purpose. The EEPROM memory space is mapped to the HI-3210 as shown in the diagram below.

All configuration memory blocks are copied. The ARINC 429 Received Data Memory contents and ARINC 429 Receive log FIFO contents are not copied to or from the EEPROM.

The HI-3210 can be used to program the Auto-Initialization EEPROM. When the HI-3210 is in its IDLE state (RUN input = "0"), a three step sequence must be performed to begin the EEPROM programming cycle:

1. Write data value 0x5A to HI-3210 memory address 0x8FFF.
2. Write data value 0xA5 to HI-3210 memory address 0x8FFF.
3. Apply a positive pulse to the PROG input pin of at least 1ms.

If the three-step sequence is interrupted by any intervening

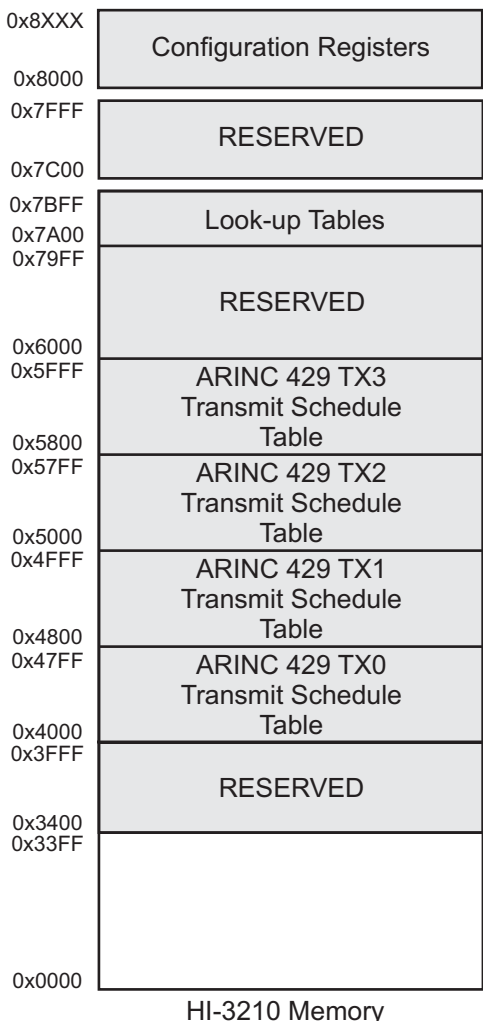
host activity between steps 1 and 2, or 2 and 3, or if the PROG pulse is less than 1 ms, the programming cycle will not start and the device remains in the IDLE state.

Taking the PROG pin low initiates the cycle. The READY pin goes low, and the contents of the HI-3210 memory and registers are copied to the EEPROM. When copying is complete, the HI-3210 executes a byte-by-byte comparison of the EEPROM and its own register / memory contents. If the verification completes successfully, the READY pin goes high.

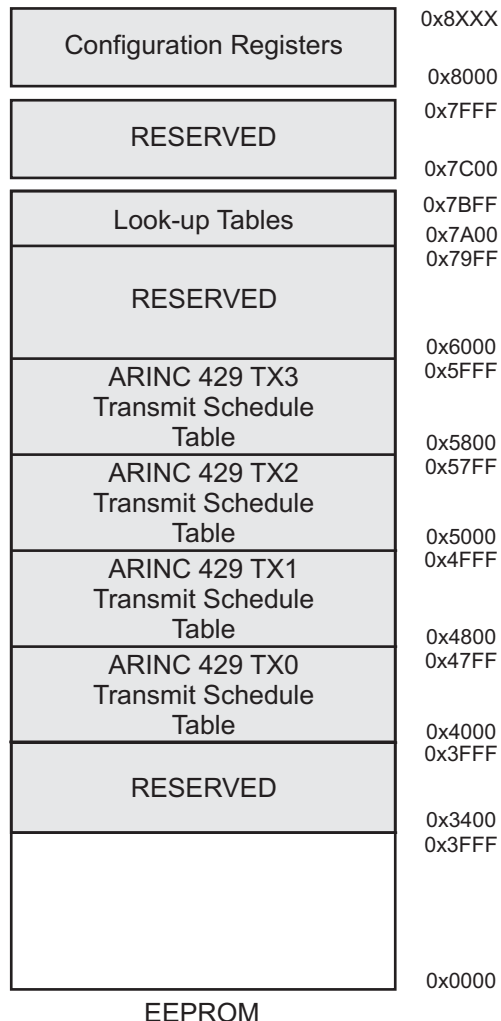
A 2's complement of the checksum is also written to the EEPROM at location 0x807F. The total read back checksum should be zero. The following locations are excluded from the checksum because they are either read-only or unused locations: 0x8000 thru 0x800e, 0x8023 thru 0x802f, 0x8036 thru 0x805e, 0x8070 thru 0x807e.

If the comparison of the EEPROM contents and HI-3210 memory / register contents results in a discrepancy, the HI-3210 enters the SAFE state, the PROGERR bit is set in the Pending Error Register and the INT output is asserted.

The user must clear the PROGERR issue before normal operation can resume.



HI-3210 Memory



EEPROM

## ABSOLUTE MAXIMUM RATINGS

Supply voltage (VDD)	-0.3 V to +5.0 V
Logic input voltage range	-0.3 V DC to +3.6 V
Power dissipation at 25°C	1.0 W
Reflow Solder Temperature	260°C
Junction Temperature	175°C
Storage Temperature	-65°C to +150°C

## RECOMMENDED CONDITIONS

Operating Supply Voltage	VDD..... 3.3 VDC ±5%
Operating Temperature Range	Industrial ..... -40°C to +85°C Extended ..... -55°C to +125°C

*NOTE: Stresses above absolute maximum ratings or outside recommended operating conditions may cause permanent damage to the device. These are stress ratings only. Operation at the limits is not recommended.*

## DC ELECTRICAL CHARACTERISTICS

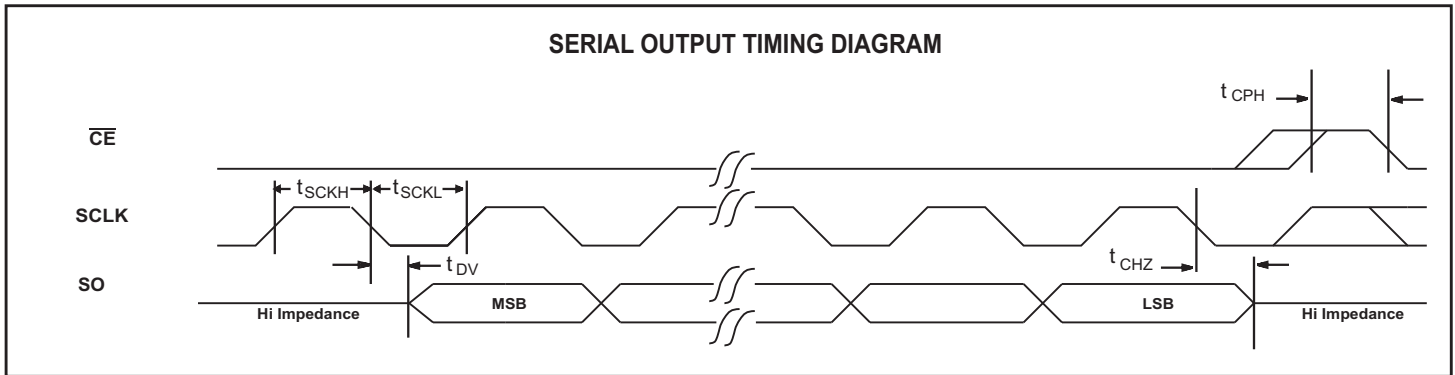
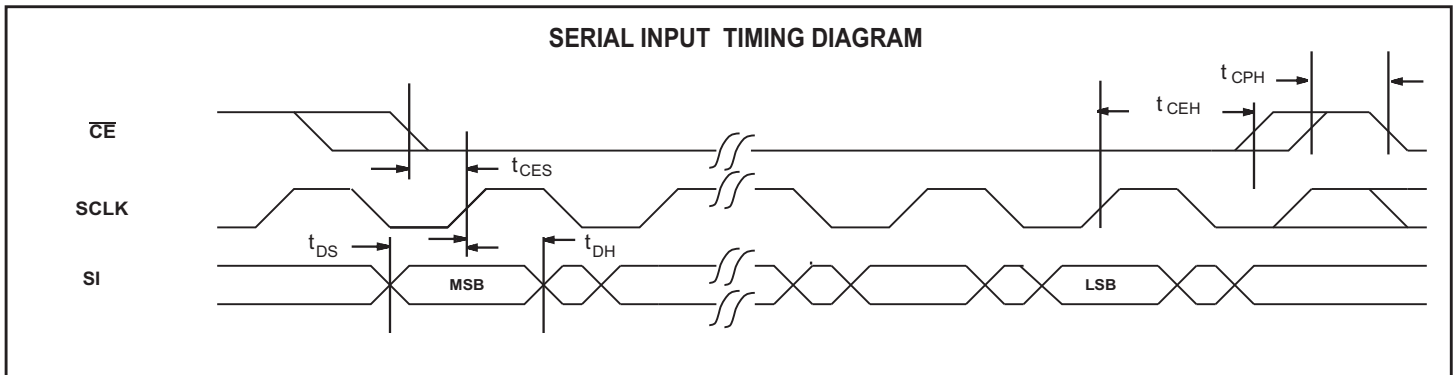
VDD = 3.3 V, GND = 0V, TA = Operating Temperature Range (unless otherwise specified).

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Operating Voltage	VDD		3.15	3.30	3.45	V
Supply Current	IDD				50	mA
Min. Input Voltage (HI)	V <sub>IH</sub>	Digital inputs	70%			VDD
Max. Input Voltage (LO)	V <sub>IL</sub>	Digital inputs			30%	VDD
Pull-Up / Pull-Down Current	I <sub>PUD</sub>	Digital inputs and data bus		30		µA
Min. Output Voltage (HI)	V <sub>OH</sub>	I <sub>OUT</sub> = -1.0mA, Digital outputs	90%			VDD
Max. Output Voltage (LO)	V <sub>OL</sub>	I <sub>OUT</sub> = 1.0mA, Digital outputs			10%	VDD

# AC ELECTRICAL CHARACTERISTICS

VDD = 3.3 V, GND = 0V, TA = Operating Temperature Range (unless otherwise specified).

PARAMETER	SYMBOL	LIMITS			UNITS
		MIN	TYP	MAX	
<b>SPI Host Bus Interface</b>					
SCK clock period	t <sub>CYC</sub>	50			ns
$\overline{CE}$ set-up time to first SCK rising edge	t <sub>CES</sub>	25			ns
$\overline{CE}$ hold time after last SCK falling edge	t <sub>CEH</sub>	25			ns
$\overline{CE}$ inactive between SPI instructions	t <sub>CPH</sub>	100			ns
SPI SI Data set-up time to SCK rising edge	t <sub>DS</sub>	10			ns
SPI SI Data hold time after SCK rising edge	t <sub>DH</sub>	10			ns
SCK high time	t <sub>SCKH</sub>	25			ns
SCK low time	t <sub>SCKL</sub>	25			ns
SO valid after SCK falling edge	t <sub>DV</sub>			20	ns
SO high-impedance after $\overline{CE}$ inactive	t <sub>CHZ</sub>			75	ns

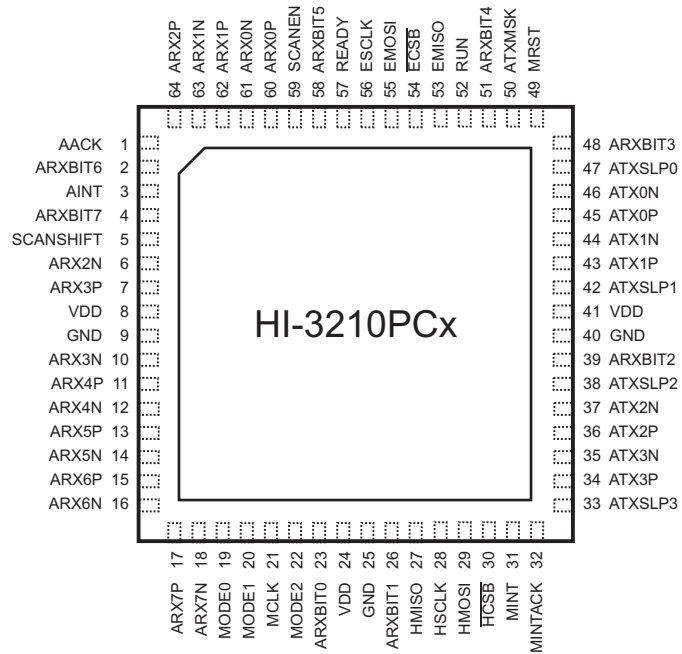




## PIN CONFIGURATION FOR HI-3210, 64-PIN QFN PACKAGE

### Notes

1. All VDD and GND pins must be connected.
2. See data sheet page 1 for HI-3210, 64-Pin PQFP Package Configuration.



**TOP VIEW**

**ORDERING INFORMATION**

HI-3210xx x x

PART NUMBER	PACKAGE DESCRIPTION
Blank	Tin / Lead (Sn / Pb) Solder
F	100% Matte Tin (Pb-free RoHS compliant)

PART NUMBER	TEMPERATURE RANGE	FLOW	BURN IN
I	-40°C TO +85°C	I	No
T	-55°C TO +125°C	T	No
M	-55°C TO +125°C	M	Yes

PART NUMBER	PACKAGE DESCRIPTION
PQ	64 PIN PLASTIC QUAD FLAT PACK PQFP (64PQTS)
PC	64-PIN PLASTIC CHIP-SCALE PACKAGE QFN (64PCS)

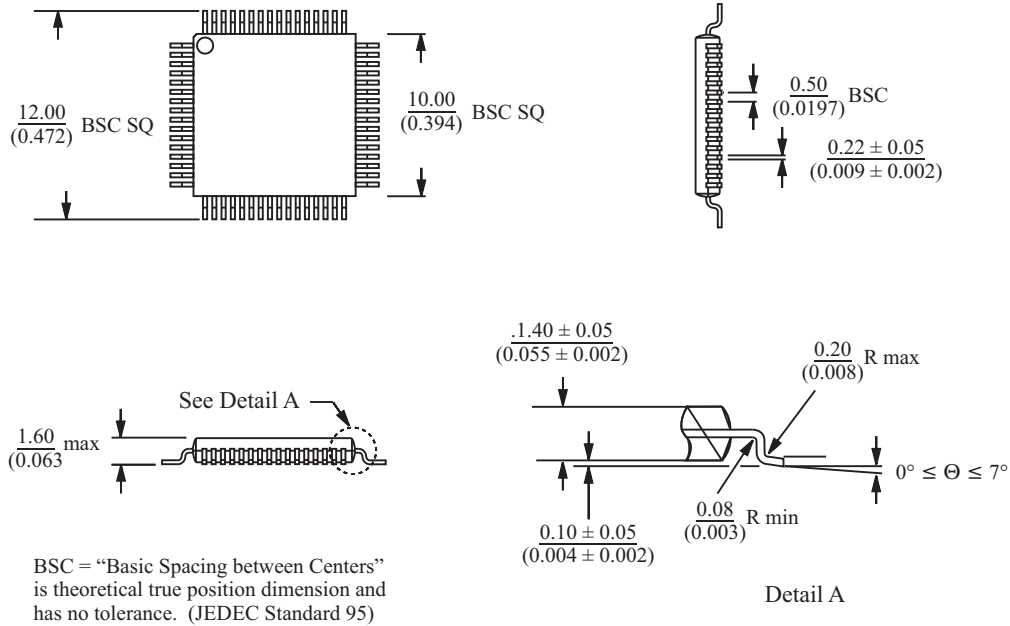
## REVISION HISTORY

Document	Rev.	Date	Description of Change
DS3210	New	5/10/11	Initial Release.
	A	1/25/12	Corrected typo in ordering information.
	B	4/16/12	Corrected fonts and missing text in numerous block diagrams.
	C	5/7/13	Added description for register 0x8035, "ARINC 429 TX Ready INT Enable". Update QFN package dimensions.
	D	8/13/13	Updated Programming for Auto-Initialization EEPROM section. Update Reflow Solder Temperature.
	E	10/22/13	Added description of ARINC 429 Receive Pending Interrupt Register, ARINC 429 Receive Interrupt Mask Register and ARINC 429 Receive Interrupt Address Registers.
	F	11/4/13	Reverse direction of signals on EEPROM in Block Diagram and Examples 4 and 5. Add "bar" (active low) to ECSB and HCSB signals in Pin Descriptions. Update PQFP-64 and QFN-64 packages.
	G	01/29/14	Clarified description of Repetition Rate Register.
	H	08/04/14	Add notes to clear RBFFAIL bit before applying RESET for modes 2 and 3. Add description of ARINC 429 Muxed FIFO Flags Register and ARINC 429 Tx Ready Bits Register.

**64 PIN PLASTIC QUAD FLAT PACK (PQFP)**

*millimeters (inches)*

Package Type: 64PQTS



**64-PIN PLASTIC CHIP-SCALE PACKAGE (QFN)**

*millimeters (inches)*

Package Type: 64PCS

